# *Definitional Question-Answering Using Trainable Text Classifiers*

**Oren Tsur**

M.S.c Thesis
Institute of Logic Language and Computation (ILLC)
University of Amsterdam

December 2003

# Abstract

Automatic *question answering* (QA) has gained increasing interest in the last few years. Question-Answering systems return an answer rather than a document. *Definitional questions* are questions such as *Who is Alexander Hamilton*? or *what are fractals*? Looking at logs of web search engines *definitional questions* occur quite frequently, suggesting it is an important type of questions. Analysis of previous work promotes the hypothesis that the use of a *text classifier* component improves performance of definitional-QA systems. This thesis serves as a proof of concept that using *trainable text classifier* improves definitional question answering. I present a naïve heuristic-based QA system, investigate two text classifiers and demonstrate how integrating the text classifiers into definitional-QA system can improve the baseline system.

**Key words**: *definitional-questions answering, information retrieval, text mining, text classification, text categorization*.

## *Table Of Contents*

## Acknowledgments

Thanks to my thesis advisors Dr. Maarten de Rijke and Dr. Khalil Sima'an for their help, support and inspiration.

I also wish to thank my supervisor Dr. Henk Zeevat for his guidance and many wise advices.

Finally, I'm happy to thank my friends at the ILLC for the wonderful time we spent together; doing their best to save me from dehydration and making Amsterdam feel like home.

# 1. Introduction

## *1.1 Question Answering*

The field of *Automatic Question-Answering* (*automatic QA* or *QA*, here after) can be viewed from many different perspectives. This introductory chapter briefly reviews the short history of the field, the contexts in which the research exists and the current research agenda. Next, I zoom-in to the more challenging task of *definition QA* in which answers are harder to retrieve and evaluate. I shall express the motivation and objectives of this work and close the introduction with a short review of the structure of the thesis.

Several disciplines are involved in *QA*, some of them interact whilst some are independent, some are of theoretical nature whereas others are very practical. The main disciplines involved are philosophy, psychology and computer science.

The roots of QA found in philosophical discussions are millennia old. Although, at first glance, it seems the issue of *questions* and *answers* is clear, the nature of ‘a question’ and the ‘expected’ answer occupied the mind of many philosophers during hundreds of years. Starting from the Socratic dialogue, knowledge, understanding, paradox, world - all define nature of “a question”. Ontology, epistemology, mind, truth, ideals, and proof - all define the nature of a good “answer”. Later on, as part of the discussion about the *evaluation problems*, we mention those philosophical issues.

Back in the 1930’s, QA became part of the psychological research as researchers were interested in the cognitive aspects of question-answering, information-need and satisfying this need. Since the 1980’s, cognitive research regained importance and popularity, and several cognitive models of QA were suggested [QUEST model by Graesser and Franklin 1990; Kuipers 1978; Daniels 1986 and more]. Looking at the psychological aspects and the cognitive models of QA can help in building QA systems and vise versa – automatic-QA system can test cognitive model and lead to new directions in the cognitive research [Dix et al. 2002; Pazzani 2001; Norgard et al. 1993 and more].

In recent decades information access has become a major issue. As processors became faster, memory and, especially, storage space became cheaper, and most of all, due to the vast growth of the Internet, we are faced with an urgent need to provide access to the available information resources in an efficient way. Document retrieval systems aim to do this by taking a number of keywords and returning a ranked list of relevant documents. Question answering systems go beyond this. In response to a question phrased in natural language, QA systems aim to return an *answer* to the user. In other words – a QA system should supply the user with only the relevant information instead of a pointer to a document that might contain this information. The user is interested in an *answer*, not in a document. The users want all of their work to be done by the machine and do not wish to do the filtering themselves. Sometimes the user is interested in an opinion and the question involves some degree of inference. The answers to this type of questions could not be obtained from a collection *as is* since the answer "as is" is not present in the collection. An understanding of the question and an inference technology should be used to process the information coded in the collection and generate an appropriate answer. The main sub fields of coputer science involved in this field of research are Information Retrieval (IR), Information Extraction (IE) and Text Mining.

As was suggested earlier, one cannot totally distinguish the philosophic-psychological aspects of QA and the practical task of automatic QA. A QA system shouldn't necessarily use a cognitive-psychology model of question-processing and answer-generation, but it should engage some knowledge about the expectations of the questioner and the context of the information-need. Moreover, one should take the obscurity of the concept of a 'good answer' into account. Although it seems that the concept of a 'good answer' is very clear, coming to a formal definition can be quite tricky. This is an acute problem especially when trying to evaluate automatic-QA systems.

## 1.2 Question Answering at TREC – Text REtrieval Conference.

Back in the 60's [Green et al. 1963] a domain-dependant QA system was built, claiming to answer simple English baseball questions about scores, teams, dates of games etc. A decade later the Lunar system was built [Woods 1977], answering questions regarding data collected by the Apollo lunar mission such as chemical data about lunar rocks and soil. The domain-dependant systems are usually based on structured databases. Those sporadic experiments didn't cause the expected research "boom" and no large-scale experiments or evaluation took place for several decades, until the first Text Retrieval Conference was held in the beginning of the 90's.

The Text REtrieval Conference (TREC), co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA), was started in 1992 as part of the TIPSTER Text program. Its purpose was to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. [NIST home page[1]; 23;24;25;26;27]. Each year's TREC cycle ends with a workshop that is a forum for participants to share their experiences. After the workshop, NIST publishes a series of overview papers and proceedings written by the participants and the TREC organizers.

| **Factoid Questions** | **Definitional Question** |
|---|---|
| • How many calories are there in a Big Mac?<br>• Who was the first American in space?<br>• Where is the Taj Mahal?<br>• How many Grand Slam titles did Bjorn Borg win? | • What are fractals?<br>• What is Ph in biology?<br>• Who is Niels Bohr?<br>• What is Bausch & Lomb? |

Table 1.1 Examples of Factiod and definitional questions

At 1999, TREC-8, *Question Answering* track was the first large-scale evaluation of domain-independent QA systems. At TREC-11 (2002) many of the questions in the test set (taken from search engine logs) turned out to be definition questions, even though the main focus of the track was still on factoids. This showed that:

[1] http://trec.nist.gov/overview.html

> "*Definition questions* occur relatively frequently in logs of search engines, suggesting they are an important type of questions" (TREC 2003, definition QA pilot; [25])

At TREC 2003 definition questions became an official part of the evaluation exercise, with their own answer format and evaluation metrics (see chapters 2 and 3 for details). To stress the importance of definition questions, they accounted for 25% of the overall QA score, although they only made up about 10% of the whole question test set.

One of the main challenges of TREC 2003 QA track was the problem of evaluation of answers. The problem of evaluation is also of great importance and I address it in more detail in chapters 2 and 3. Evaluation of QA systems means a clear idea of what a good answer is. As mentioned above, this problem is not only a computational problem (as hard as the answer retrieval itself), but it is also an old philosophical and psychological problem. My interest in building a QA system is motivated not only by achieving another step for a novel solution to the QA problem, but it is motivated also by those philosophical and cognitive questions. Note that the TREC evaluation is still done by human assessors while the main effort was defining metrics and guidelines for the assessors as a starting point before building an automated evaluation system.

## *1.3 Objectives and Structure of the Thesis*

The previous sections presented the TREC research agenda and the different research possibilities. In this section I present my interests and my research agenda, entwined with the TREC agenda in some aspects and differs in other aspects.

> In this work I'm concerned with ***open domain Definition QA***. My interest lies in ***open corpus*** source, namely the ***WWW***. The WWW presents the research community a great challenge with benefits on top. Unlike other collections and databases, nowadays, the web is accessible to everyone. There is an incredible wealth of information on the web, implying an answer

can be found to almost any possible question. The web is changing constantly and new emerging events have their reflection on the web. On the other hand, it is unstructured, constantly changing, not supervised and contains much noise and low quality information. My challenge is to cope with this wild field of data in order to mine the gold-lines of information that lies beneath the messy surface.

Next, in **chapter 2**, I present the state of the art and some of the core challenges we face when we come to deal with ***definitional*** *QA* systems. Amongst those challenges I discuss the evaluation problem, the "good answer" definition problem, the recent TREC guidelines for answers and assessment and I state my own objectives.  I also mention some problems regarding web retrieval.

In **chapter 3** I present my baseline definition-QA system. This system is based on a heuristic selection of feature-sets and keywords for a good retrieval. This system was submitted to the TREC 2003 QA track and was ranked the seventh out of 54 submissions by leading corporations and institutes. I'll present a differently tuned version of the baseline that performed even better. I'll analyze the results, the points of strength and the weakness of this baseline system. I conclude that we can improve performance using *machine learning* techniques for *text categorization*.

**Chapter 4** is an introduction to *text categorization*. I describe the classification problem from general perspective, discuss some crucial aspects of text classification algorithms and explain why this problem is not linearly-separable. I also briefly review the two classification algorithms I use –a naïve classifier - mutation of RIPPER and Support Vector Machines (SVM). These two algorithms were chosen for they represent two different approaches toward text classification, each has its unique advantages and disadvantages.

**In chapter 5** I present a version of the RIPPER classifier I built specifically for this categorization task. I review and discuss my implementation and results.

**Chapter 6** repeats the categorization with another algorithm – SVM. It is assumed that SVM is currently the best text classifier. I present and discuss some of the surprising results.

**Chapter 7** goes back to the over-all picture presenting the unified system, tested on a real definitional QA task, and analyzes the results.

**Chapter 8** closes this work with conclusions, future work and further research to be done.

# 2. Definitional QA – Background and Challenges

This chapter presents the main challenges and problems of definitional QA, the state of the art of definitional QA and the guidelines of the last TREC QA Track, which define the current research agenda in the field. I review a few problems regarding QA and web retrieval and mention some of my goals and objectives beyond the TREC guidelines.

Fact -Q: Where is the Taj Mahal?

A: In the city of Agra.

Def –Q: What is a battery?

A: Batteries are made in cell packs and button-size disks, and last from days to months and even years. All modern batteries generate electricity by use of an electrochemical reaction. Generally two electrodes are inserted--one positive (a cathode) and one negative (an anode)--into a material called an electrolyte, which helps the flow of energy between electrodes. The electrodes can be liquid or solid. When a battery runs down the anode and cathode reach a state where they can no longer pass electrons between them. Batteries store energy not electricity. Batteries date back to 1790. Zinc carbon batteries have given away to Alkaline batteries. Over the years batteries have used manganese dioxide, zinc, nickel, lithium salts, hydrogen-absorbing alloys and potassium hydroxide.

Table 2.1 Example of answers for factoid and definition questions (TREC documents).

## 2.1 Characteristics of the Definition QA

Definition questions, as they are used at TREC and by most of the researchers, opposed to factoid questions, has two main features (see table 2.1)[27]:

1. The answer is longer and consists of at least one full sentence and can be of a paragraph length or even longer.
2. The correctness of the answer is not binary. The answer *A* for a factoid-question *Q* is either correct or incorrect[2], while the answer *A'* for a definition-

[2] Actually there also is "inexact answer" – "the string contains more than the answer or missing bits of the answer" [TREC 2003 guidelines]. This "inexact" refers to a string that obviously has the answer but contains an additional noise. In the factiod QA track the guidelines are to get the cleanest short answer. In the context of factoid questions, "Inexact" is a very strict notion referring to the answer string and not to the answer itself.

question *Q'* can be partially correct, partially wrong, not a full answer, inexact answer etc. Using terms of IR, we have to define some metric of *precision* and *recall* in order to rank answers for definition-questions.

Looking at table 2.1, the answer to the factoid question is correct or incorrect, although it can be incorrect for number of reasons. 'Agra' is the right answer while in case the system returns "Jerusalem" or "Amsterdam" it would clearly be a wrong answer, which no one can argue about. The full answer is a string of 5 words and the answer could also be a single word "Agra" or two words "Agra, India"[3]. The Answer for the definition-question is much longer, based on many facts from different areas, requires more sophisticated language model and text generation mechanism. Moreover, is it a full answer or maybe it is too long and detailed? Maybe the question was about an artillery battery and not about the electricity power source?

The vague nature of an answer to a definition-question also results in a major difficulty to evaluate the answer. Automatic evaluation of a definitional-QA system is as hard and problematic as the answer retrieval itself or even harder. In order to automatically evaluate an answer, there should be a clear notion of what a good answer is. Furthermore, the evaluation system needs to occupy a good comparison method in order to compare the information the system expects, and the information retrieved, since even though good answers deliver close information, the language used is probably different. In other words, the system should be capable to match two semantically-close answers although they differ very much in style. In addition, we totally ignore the fact that different users expect different answers and one user can find an answer sufficient while another finds it insufficient. In an ideal world we would like the evaluation system to be flexible to users need.

All of those problems were addressed in TREC 2003 [27;28] but no sufficient solution yet found, therefore the current TREC policy is to evaluate definition-QA systems by human assessors. Human assessors have the cognitive ability to evaluate

[3] Actually it is not that trivial. Should we accept an answer like "India"? Should we accept an answer like "Asia"? what about the given answer "The city of Agra" – shouldn't we require the state as well? What about the answer "On the banks of Yamuna river"?
However, we deal with definition questions and not factoid questions. All these problems and vagueness of factoid questions are much more acute and inherent in definition questions. A vast discussion about factoid questions can be found in previous TRECs publications.

the answers but the core problem still remains – what is a good answer. For a specific question *Q*, one will take $A_1$ as a sufficient answer while another will take $A_2$ as a sufficient answer. $A_1$ and $A_2$ can have almost no intersection at all[4]. Different assessors, just like different users, have different expectations for answers. In the battery example (table 2.1) one will be much more interested in the chemical reaction while another is interested in the different sizes and uses.

Not only that those differences of "expectation" exist among assessors, each assessor might change his point of view and way of judging in different circles of evaluation. In the last TREC 2003 it was shown that even the same assessor could evaluate a specific answer in more than one way in different rounds of assessment [10;27][5].

Another characteristic of the nature of the answer of a definition question is its eclecticism. The answer consists of many pieces of information, sometimes from different sources and different fields. Looking back on the battery example, we can find historic facts, chemical facts, electronic facts and something about the way batteries are used. Although not yet required by TREC, ideally, all those facts should be presented in a coherent and readable way, and not only as separate nuggets. In case the information was extracted from different sources, it should loose the signs of the different origin (anaphora etc.) and be a part of the newly generated text – the final answer (see appendix B) [27;28].

*Automatic text summarization* is another interesting and very challenging task sharing some important characteristics with definition-QA. Text summarization is the process of distilling the most important information from a source (or sources, in the case of multi-document summarization) to produce an abridged version for a particular user and task [16;18;19]. Identifying the key points in a document or documents extracting them and organizing/rewriting the extracted snippets in a coherent way. Automatic text summarization research is still an emerging field, but the current definitional QA research can borrow some of the insights, methods and metrics used for text

[4] The notion of "intersection" of answers in not well defined yet. I define it later.

[5] Each group participating in the TREC could send 3 different runs of their system. Even though some of our runs were identical, they were evaluated differently.

summarization. Indeed, some of those metrics were used by the TREC 2003 assessors as reviewed later in this chapter.

## 2.2 State of the Art

As mentioned above, TREC 2003 was the first large-scale effort to evaluate definitional QA systems. Not much research on definitional QA took place before this last TREC. AskJeeves[6], probably the most known QA system is incapable of answering definitional QA due to the simple fact that it retrieves a list of documents, just like a standard search engine. Other systems like AnswerBus[7] [Zhiping Zheng] retrieve a series of snippets, most of them irrelevant or insufficient. TREC committee implemented and tested few systems as a pilot to the real track. These systems were tested in order to set the goals for the coming TREC (2003). This section briefly reviews two of the very few definitional QA systems were built before TREC 2003.

### 2.2.1 *Google* Glossary

Google Glossary is not a real QA system but it shares some important aspects with QA systems, therefore I find it worth mentioning. Last summer (July), just before the TREC deadline, Google released a beta version titled *Google Glossary*[8]. Google released no official reports about the system performance or about the algorithm. Google declares the system "Finds definitions for words, phrases and acronyms" [Google Glossary]. It seems Google Glossary exploits few online glossaries and encyclopedias and retrieves a list of the definitions taken from those glossaries. It seems that Google Glossary is searching for the definition target in *external knowledge source* sources. As for its domain, it retrieves definitions for words and phrases but not organizations or figures, unless they are very well known and won an entry in one of the encyclopedias accessed by Google[9].

[6] http://www.ask.com/

[7] http://misshoover.si.umich.edu/~zzheng/qa-new/

[8] http://labs.google.com/glossary

[9] An "exception" for a definition question of an organization answered by Google Glossary is for the term "Yahoo". "What is Yahoo?" is a question taken from TREC 2003 pilot. Google Glossary retrieves the answer "The most popular and (perhaps) the most comprehensive of all search databases on the World Wide Web. Yahoo's URL is http://www.yahoo.com"
The full list of answers can be found at http://labs.google.com/glossary?q=yahoo.

Opposed to a real QA systems, Google Glossary lacks the ability to process a natural language question. All *wh* questions cannot be answered since Google Glossary fails to process it. Google Glossary requires the bare term (definition target), while the main idea of QA systems is to process a natural language style question and return a good answer in natural language. “Search engines do not speak your language, they make you speak their language” [AskJeeves]. QA systems attempt to let the users ask the question in a way they would normally ask it. In those terms, Google Glossary is still a standard search engine.

### 2.2.2 DefScriber

DefScriber is a system developed in Columbia University. It combines knowledge-based and statistical methods in forming a multi-sentence answers to definitional questions of the form “What is X”. It is based on definitional predicates by which it identifies the “right” definition sentences from a selection of possible answers. The candidate sentences are selected from longer documents by summarization and generation techniques. The main idea of the system is that “answering a ‘What is X?’ definitional question and creating a summary result for the search term ‘X’ are strongly related” [6]. The predicates mean to screen definitional material from a large volume of non-definitional material returned from search and identifying types of information typically found in definitions. Detailed description of the system architecture can be found in [6].

The idea of identifying the definitional material from non-definitional material is very appealing and indeed one of the challenges of this thesis is to try to identify the *definitional material* using ML techniques (see chapter 4). However, the weakness of the DefScriber system is the predicates dependency. The system searches a document and identifies 3 pre-defined predicates. Not only that 3 is not sufficient number of predicates, but the need to manually choose the predicates is not noble. Defining the right predicates is a delicate task that has to be done for every type of definition, which requires a lot of (human) effort. In the baseline implementation I sent to the TREC, I used a somewhat similar approach, using feature vector and features selection instead of predicates (see chapter 3). Very good results were obtained this way but analysis of the results shows that better results can be achieved by using ML techniques instead of heuristics and manual feature-selection.

## 2.3 Official TREC Guidelines and the Evaluation Problem

Evaluation of definitional QA systems is much more difficult than evaluation of factoid QA systems since it is no more useful to judge system response simply as right or wrong [see section 2.1; 27; 28]. Evaluation of definitional QA system is actually abstraction of the real system's task of providing an answer to the definition question. This is just like the chicken-and-egg proposition, since the whole point of evaluation is to set the standards that constitute a "good system".

The last TREC set the requirements from definition-QA systems, as currently expected in this relatively new field of research. Traditionally, the TREC committee sets standards and milestones for improvement before and after analyzing the conference results. In the pilot of the last definition-QA track, "systems were to return a list of text snippets per question such that each item in the list was a facet of the definition target. There were no limits placed on either the length of an individual nugget or on the number of nuggets in the list, though systems knew they would be penalized for retrieving extraneous information" [27]. Yet, assigning partial credit to a QA system response requires some mechanism for matching concepts is the desired response to the concepts in the system's response, similarly to issues arise in evaluation of *automatic summarization* and *machine translation* systems, where the concepts should be semantically identical and not syntactically identical. Moreover, without some expectations and "familiarity" with the user (questioner) it is impossible for the system to decide what level of detail is required and considered a "good" answer. School-aged child and a nuclear scientist expect two different answers for the same question. To provide guidelines to systems developers, the following scenario was assumed:

> The questioner is adult, a native speaker of English, and an average reader of US newspapers. In reading an article, the user has come across a term that he would like to find more about. The user may have some basic idea of what the term means either from the context of the article or basic background knowledge.

> They are not experts in the domain of the target, and therefore not seeking esoteric details [27].

Taking the above into account, precision and recall are non-trivial notions when scoring definition questions (or text summarization). The TREC committee solved the problem as follows: "For each definition question the assessor will create a list of all acceptable information nuggets about the target from the union of the returned responses and the information discovered during development. Some of the nuggets will be deemed essential, i.e. piece of information that must be in the definition of the target for the definition to be considered good. The remaining nuggets on the list are acceptable: a good definition of the target may include these items but it doesn't need to" [27; 28].

The definition of precision and recall is derived from the assessor's lists, although an answer could achieve a perfect recall by matching only a subset of the list – all the essentials. On the other hand, there are many other facts related to the target that, while true, detract from a good definition and thus an answer containing those facts is penalized. These facts shouldn't appear in the assessor's list[10]. The scoring metric for definition answers is based on nuggets recall (*NR*) and an ***approximation*** of the nuggets precision (*NP*) based on length, assuming that of two strings that deliver the same information but vary in length – the shorter the better. The *NP* and *NR* are combined by calculating the F-score with β = 5 to give recall an importance-factor of 5 comparing to the precision.

The F-score measure is computed as follows:

$$F_\beta(L_Q^i) = \frac{(\beta^2+1)NP \cdot NR}{\beta^2 NP + NR},$$

where $L_Q^i$ is a list of snippets (pieces of information) retrieved by system $i$ for a question *Q*. *NP* and *NR* are standard precision and recall computed against $L_Q^*$ - the assessor's target list (for more details on the evaluation metric see section 3.3).

[10] In a question about Alger Hiss, we don't care about the fact that he brushes his teeth every morning, even though it may be true.

Note that although this is a formal and strict metric, it only addresses some of the aspects mentioned above. An assessor still has the some freedom to create his own list and to decide which facts are essential and which are not – this way the assessor affects the recall and the F-score of an answer. A good question-set is of a diverse nature and as the assessor is not an expert in the fields of the entire collection, he learns the answer just like the user and it is possible that his lists are not complete. Moreover, different assessors, just like different users, can legitimately have different views on what a relevant or essential snippet is.

Another issue the TREC F-score metric completely ignores is the coherence and organization of the answer. This problem was tackled in the TREC pilot "Holistic" evaluation. In this evaluation the was score computed as follows:

$$score(L_Q^i) = 5 \cdot content + \frac{1}{2} content \cdot organization$$

This is very loosely defined. The assessors were to judge intuitively given the definitions (TREC 2003 definition QA Pilot):

> *Content*: The system response includes easily identifiable information that answers the question adequately; penalty for misleading information.
>
> *Organization*: Information is well structured, with important information up front and no or little irrelevant material[11].

This metric cannot support a good quantitative evaluation for it is too loosely defined. The definition of organization is wide open for self-interpretation of the assessors, moreover, the pilot systems scored poorly in the organization axis and the requirement for organization was postponed to future TRECs. At this phase of initial research of the field, the organization bar seems slightly too high for both- system performance and evaluation. However, this thesis aims to achieve organization as well as precision and recall. The ambition to treat organization as well as precision and recall results in the methods I use to retrieve my answers. In the next chapter I present two versions of the baseline system. The first version is the version submitted to TREC 2003, the other is a differently tuned version of the TREC submission. Evaluation of both versions is based on the TREC F-score metric used by TREC

[11] "Irrelevant material" means true facts connected with the target but irrelevant for its definition.

assessors. In appendix B there is an alternative metric, little different  from the one proposed in the TREC pilot.

## *2.4 The Corpus*

### *2.4.1 Which Corpus to use?*

TREC QA track is based on a closed corpus. Each system submitted to the TREC can get its answers using all sources available but the answers should be justified by pointers to the documents in TREC collection from whom the answer could be retrieved. My definition QA system is based on *web retrieval* and not on the closed corpus domain used for the TREC QA track. Basing the system on web retrieval forced me to add another component to my system – the justification component. This component was used next to the web retrieval in order to find "justification" to my web retrieved answers in the TREC collection. This component is irrelevant for this research and was used only in order to meet the TREC guidelines. Yet, using web retrieval involves few other issues I find important to mention.

The most important feature of the web is its size. The web is huge and dynamic. Information can be found on almost every desired topic. For instance, on November 24, Google has indexed 3,307,998,701 pages, which is only a portion of the actual web.

On doing web retrieval, this wealth of information means that recall is not an issue and only precision counts since full recall cannot be obtained and/or computed. One can hardly tell whether he has the full picture due to the size and the constant change of the web content represented (in this research) by the Google indexes[12]. On the other hand, doing definitional QA, recall is an important measure to the extensiveness of the answer. Definitional QA based on web retrieval is, therefore, problematic.

This problem was artificially solved by adopting the F-score or any similar metric basing precision and recall on finite, relatively short lists. The assessor decides on a list of facts that are acceptable as part of an answer. A subset of those facts is defined

[12] All the systems implemented this thesis make use of Google search engine as a preliminary retrieval engine.

as mandatory essential fact that no answer could be considered complete without. Precision and recall are then computed in respect to those lists and not in respect to the information “available” on the web. Note that this solution is only technical and not inherent. Those lists enable to measure precision and recall (and thus F score) in respect to those lists only and give a numeric evaluation, but one can argue that the lists are not complete unless built by a universally-accepted expert in the field of the definition target.

### *2.4.2 Web Retrieval – The Truth is Somewhere Out There*

Another issue regarding web retrieval is the quality of the pages. One of the things that characterize the web is its lack of regulations. The web is in a constant change; there is no control on the contents – subject wise and quality wise. It is not guaranteed that a document or a nugget retrieved from the web is not noisy or misleading. One can assume that Google page ranking system overcomes this problem. This assumption is only partially true. While it is reasonable to assume that high-ranked Google pages are more important *hubs* and *authorities*, contain less junk and are better structured, these pages can still be biased and contain misleading information. For example, Microsoft PR page is the first to be retrieved for some queries regarding Microsoft Corporation, Microsoft products or Microsoft personnel but one has to remember that Microsoft Corporation dominates the Microsoft domain with all the information bias derives from the interests of the “page master”. This is not some Microsoft conspiracy – it holds true for governmental domains as well as many other firms and corporations. It is even legitimate or at least a common PR tactic. Yet, we/the users have to be aware of those biases. Google ranking helps a lot to filter the real junky or poorly structured stuff out but it cannot secure objectivism of important pages. This objectivism is most important in definition questions for a definition can, sometimes, depend on the definer authority [12].

In this work I didn’t try to tackle this problem of the quality of webpages, I simply rely on the Google ranking system as the best free filtering system available.

After introducing the challenges involved in definitional QA answering and answers evaluation, presenting the research agenda defined by NIST in TREC, explaining the decision to use the web as the system corpus and presenting few issues regarding to this decision, I shall now present the baseline implementation that was sent to TREC 2003.

# 3. The Baseline QA System – TREC Version

The previous chapter drew the challenges of definitional QA and the current research agenda. This chapter describes the baseline system designed according this agenda. The baseline system was submitted to TREC 2003 and scored very nicly, ranked 7th out of 54 submissions. In this chapter I present my hypothesis, describe the system architecture, present, discuss and analyze the results the baseline system achieved in the TREC.

Along with the TREC submission results I present results of the same baseline system, a bit differently tuned. It is shown that tuning results in great improvement and I believe that the tuned run could score even better if it had been submitted to the TREC.

I close this chapter with some conclusions suggesting that using ML techniques for *text categorization* could result in further improvement.

## *3.1 Hypothesis*

Previous work showed that using glossaries improve definitional QA [7; 17]. Chu-Carroll and Prager use WordNet for "what is" questions [7]. Our strategy is that this approach can be adapted to other types of definitional questions such as "who is/was" and "what is <an organization>". In this baseline implementation I adopt this approach in two levels. The basic level is exploiting biographies collections as it was done in the past with term-glossaries by Google in their newly released beta version for Google Glossary (see section 2.2.1) and by Chu-Carrol [7] and others. In the other level I try to identify relevant pages - pages that are not part of *external knowledge source*s such as biographies collection, encyclopedias, terms-glossaries or dictionaries. This way we look at the Internet as a glossary (or biography collection, or encyclopedia) with noise.

I expect that identifying the “right” documents on the web will not only improve the precision and recall obtained by a QA system, but will also result in major improvement in the coherence and organization of the answer. The *right document* means retrieving glossary-entry-like or encyclopedic-entry-like pages from the web instead of using an *external knowledge source*, and using Google snippets representing this page as the answer nuggets.

## 3.2 System Overview and System Architecture

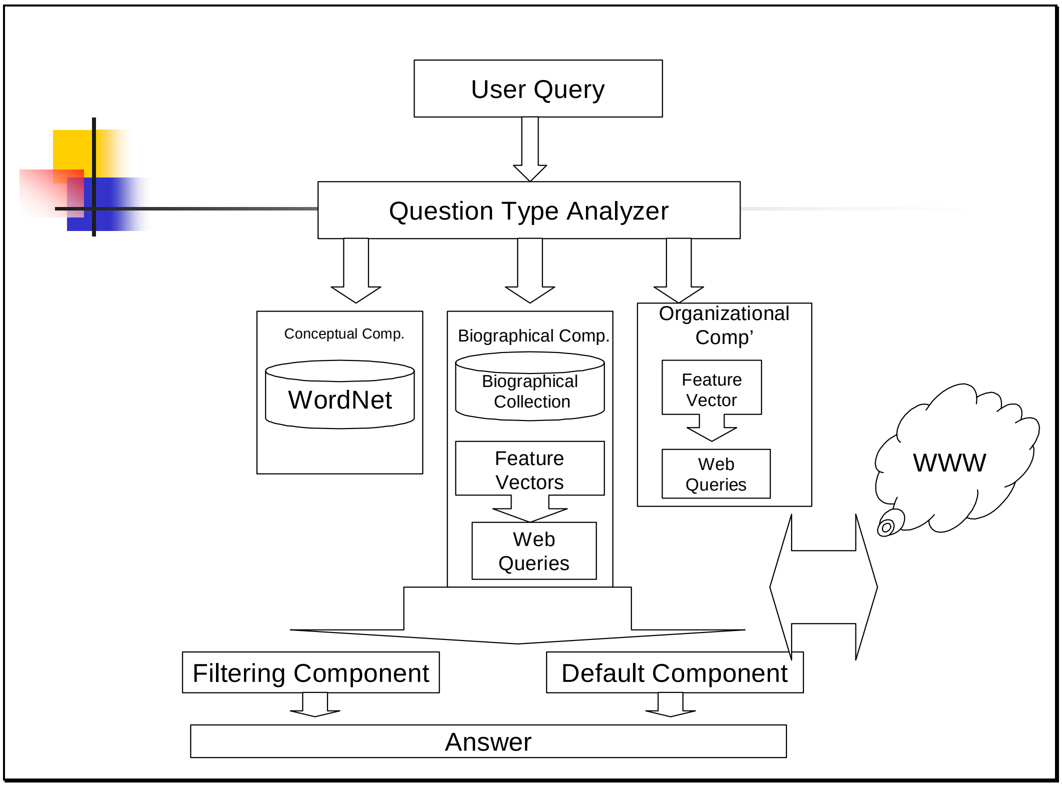

Table 3.1 Baseline System Architecture

The flow of the baseline system consists of four steps:

1. Question Analysis.
2. Answers retrieval.
3. Answers filtering.
4. TREC adjustment.

### 3.2.1 Question Analyzer

In the first step the system uses the *Question Analyzer* component in order to determine the question type (biographical, concept, organization) and key words like the question word, the definition target (DT) etc. The analysis is rather naïve and based on the question key words, like “Who/what”, on articles and determiners identification, on capitalization and name recognition.

In the next step each question type is treated in a different way by a different component. In case no answer was returned by any task-specific component, the *default component* returns a naïve answer.

### 3.2.2 Conceptual Component

Concept questions are treated by the *Conceptual component*. Previous work [7; 17], shows that using dictionary sources is very effective approach for answering “what is” questions. Given a question the system first consults WordNet in order to get the answer for “free”. This approach was recently adapted by Google, releasing its beta version of Google Glossary (see section 2.2.1).

### 3.2.3 Biographical Component

Designing the baseline system, the main effort was focused on the “Who is/was” questions. I chose to focus on “who is” questions for three reasons:

1. Analysis of search engines logs show that most of the definitional questions are of the “who is” type – about sixty (60%) percents [26; 28].
2. Answers for “who is” questions are more intuitive to understand than answers to “what is” questions since an answer to “who is” question is more likely to be a list of facts connected in a coherent way. This characteristic of the “who is” question puts it closer to the TREC “nuggets” requirements. Moreover, for this reason, biographical features are easier to “guess” and use as parts of a heuristic method (the baseline) with no crucial need for domain expert. Furthermore, biographical facts seem to be an interesting case for *text categorization* research (see chapters 4,5,6), therefore “who is” questions can serve as a case study to the other types of questions.

3. Previous work already addressed “what is” questions. Although this work suggests a new approach that wasn’t tried for “what is” questions either, the baseline, presented in this chapter, starts up with almost the same approach of exploiting existing glossaries and ordered collections, more specific, by making use of biography collections.

Focusing on “who is” questions doesn’t mean other types of definitional questions left aside. It only means I put more effort in answering “who is” questions and that the hypothesis will be checked mainly according to the system performance on this type of questions.

Biographical “Who is” questions are passed to the *Biographical component*. Following the same rationale of using WordNet for conceptual questions, we first try to get a ‘ready made’ authorized biography of the definition target (the person in question). We do this by consulting the big collection of biography.com. The challenge lies in the case where no biography could be found in the biography.com domain. Actually this happens in most of the cases and the system has to collect small biographical nuggets from various sources. The system uses a predefined-hand crafted set of human features: the *FV* (features vector). The FV contains words or predicates like “born”, “died”, “graduated”, “suffered” etc. then searches Google for different combinations of the definition target and various subsets of FV.

| | |
|---|---|
| 1 | He **wrote** many short stories, including "The Man Without a Country" 1863, "The Brick Moon" 1869 (1st story describing an **...** **Hale**, **John** Rigby, **Sir** (1923-1999 **...** |
| 2 | Editorial Reviews Synopsis **Sir John Hale** is one of the worlds foremost Renaissance historians whose **book** "The Civilization of Europe in the Renaissance" (1993 **...** |
| 3 | **...** he died there on Christmas Day, 1909 **SIR JOHN** RIGBY **HALE** **...** on the founding of Virginia,and **wrote** short stories. The grand-nephew of Nathan **Hale**, in 1903 he was **...** |
| 4 | **...** Am looking for possibly James **HALE** who was a coachman **...** for a complete list of the music WH **wrote**. **...** du nouveau manuel complet d'astronomie de **sir John** fW Herschel **...** |
| 5 | A love unspoken Sheila Hale never thought she was worthy of her husband, the brilliant historian, **Sir John Hale**. But when he **suffered** |
| 6 | On 29 July 1992, Professor **Sir John Hale** woke up, had **...** For her birthday in 1996, he **wrote** on the **...** **John Hale died** in his sleep - possibly following another stroke **...** |
| 7 | Observer On 29 July 1992, Professor **Sir John Hale** woke up **...** her birthday in 1996, he **wrote** on the **...** **John Hale died** in his sleep - possibly following another stroke **...** |
| 8 | **...** **Sir** Malcolm Bradbury (writer/teacher) -- Dead. Heart trouble. **...** Heywood **Hale** Broun (commentator, writer) -- Dead. **...** **John** Brunner (author) -- Dead. Stroke. **...** Description: Debunks the rumor of his death. |

Table 3.2 Example of Google Snippets retrieved for subsets of a toy FV and the pilot question “Who is Sir John Hale?”

The FV subsets are predefined by humans according to thumb-rules, common sense heuristics and as a result of try-and-error runs (ideally, those features would be selected by a domain expert). An example of a toy FV is be *“born, position, wrote, book, work, prize, known, graduated, school, suffered, died”*. Taking two subsets of this vector, say, “*born wrote graduated*” and “*wrote suffered died*” with the definition target, say, “Sir John Hale”[13], creating two Google queries returns various snippets some of them contain usful information and other snippets are irrelevant (see table 3.2).

The features subset selection can be much improved with experience gained over time. The features-subsets-selection is part of the parameter tuning to be described later (along with tuning of the *filtering component*).

### 3.2.4 Organizational Component

“What is <an organization>” questions are processed by the *Organizational component*. The organizational component is based on rather naïve method. It looks for information about the organization by performing a web search combining the definition target and a small subset of “organization features” that might happen to be unique to organizations manifesto or declaration. Opposed to the *Biographical Component*, the *Organizational Component* has no smart feature selection and the answer is not chosen to be the combination of different features subsets queries.

### 3.2.5 Default Component

In case the task-specific components yield no satisfactory results, the question is processed by the *Default Component*. The default component basically gets the Google snippets corresponding to the fixed string “*DT* is a”. This naïve method

[13] This was one of the most difficult questions in the definitional QA pilot. Not much information can be found on the web and most of the pages contain other people’s biographies as well so the Google retrieved snippets are quite noisy. The best online source is http://members.aol.com/haleroots/famous.html containing short biographies of “famous Hales”.

proved to be very good for questions like “2385. what is the Kama Sutra?”. The problem with this component is that there is no control on the results and no corrupted snippet will is filtered out. The use of this component depends on the penalty the system gets for wrong answers.

### 3.2.6 Snippets Filtering Component

When the system processes a question using other component than the default component it monitors the quality of the snippets. There is more than one way the retrieved snippets may be corrupted. Snippets can be noisy (see snippets 1, 3, 5 in table 3.2) or totally irrelevant (see snippets 4,8 in table 3.2). In order to overcome the problem of the lack of integrity of the snippets, we use the *Snippets-Filtering Component* to be described below.

*Snippets Filtering component* – the final step is to filter the retrieved snippets. The retrieval components might return hundreds of snippets, some are identical, some are different but contain almost identical information (see snippets 6,7 in table 3.2), some are completely irrelevant (see snippets 4,8 in table 3.2), some are relevant but dirty (see snippets 1, 3, 5 in table 3.2) while some other snippets are just perfect (snippet 2 in table 3.2 is almost perfect). Filtering the valuable snippets from the junk and identifying snippets that contains semantically close information are important tasks. These two tasks are processed separately. First we remove irrelevant snippets. An irrelevant snippet is identified by the syntactic structure of the snippet, meaning the distances between the tokens of the definition target and between the definition target and other entities and features in the snippet. In the example of table 3.2, snippets 4 and 8 are being removed for the words “Sir”, “John”, “Hale” and “wrote” are too far from each other, suggesting the snippet’s subject is not “Sir John Hale” but represents a document regarding at least three different people: “Sir X” “John Y” and “Z Hale”.

In order to remove semantically close snippets we define a *snippets-similarity metric*. The metric is loosely based on the *Levenshtein Distance (LD)* measure of similarity with some improvements to adjust it to the distance between concepts instead of strings.

Formally*, LD* is a measure of the similarity between two strings, which we will refer to as the source string *s* and the target string *t*. The distance is the number of deletions, insertions, or substitutions required to transform *s* into *t*.

The system performs a partial stop-words removal and stemming, then sorts the tokens of the each snippet in order to overcome structure variations like passive/active forms. Going back to the example presented in table 3.2, snippet number 7 is removed because its semantic similarity with snippet number 6.

| | |
|---|---|
| 1 | He **wrote** many short stories, including "The Man Without a Country" 1863, "The Brick Moon" 1869 (1st story describing an **...** **Hale**, **John** Rigby, **Sir** (1923-1999 **...** |
| 2 | Editorial Reviews Synopsis **Sir John Hale** is one of the worlds foremost Renaissance historians whose **book** "The Civilization of Europe in the Renaissance" (1993 **...** |
| 3 | **..** he died there on Christmas Day, 1909 **SIR JOHN** RIGBY **HALE** **...** on the founding of Virginia,and **wrote** short stories. The grand-nephew of Nathan **Hale**, in 1903 he was **...** |
| 5 | A love unspoken Sheila Hale never thought she was worthy of her husband, the brilliant historian, **Sir John Hale**. But when he **suffered** |
| 6 | On 29 July 1992, Professor **Sir John Hale** woke up, had **...** For her birthday in 1996, he **wrote** on the **...** **John Hale died** in his sleep – possibly following another stroke **...** |

Table 3.3. The snippets actually returned by the baseline system

The delicate task is choosing threshold of similarity, meaning the maximal distance in which we still treat two strings as semantically identical[14]. This threshold has a significant impact on the filtering task, and tuning the threshold after the track submission resulted in great improvement of precision and recall, not to mention the improvement in length of the returned answer, for many snippets were filtered out. The threshold was determined manually after several runs on a test collection and on the TREC pilot collection. Back to the example of table 3.2, setting the threshold too low might leave both strings (6, 7) in the final answer. Setting the threshold extremely high might filter all snippets but the shortest one.

In the TREC submission the system used another component – the *Justification Component*. The TREC guidelines demand an answer justification pointing to the relevant documents in the TREC collection. Since this research concerns an open corpus, namely the web, pointers to the relevant TREC documents had to be found as

[14] By the term semantically identical I don't mean *pure* semantically identical. Two strings differ from each other by a negation word will be treated as semantically identical by the system. The system 'assumes" that if the LD between two strings (stemmed, naked of stopwords and lexically ordered) is small enough, the two strings deliver the same information.

a last step. The *Justification Component* finds the justification in the TREC collection and adjusts the output to the TREC guidelines, meaning that sometimes we had to break a perfect answer to shorter nuggets. Since this work addresses the problem of definitional QA using an *open corpus*, I'm not going into the details of this component.

### 3.3 Evaluation Metric

Chapter 2 section 2.3 presented the evaluation problem and briefly described TREC 2003 metric used in the evaluation of definitional questions. This section goes into the details of the evaluation metric and assessors guidelines.

<table>
<tr><td colspan="4"><b>Hale, John Rigby, Sir</b> (1923-1999) British historian; chairman of National Gallery 1974-1980; wrote "The Civilization of Europe in the Renaissance" (1993). Sir Hale is known as one of the foremost renaissance historians and his book considered the best book ever written about those times. At the age of 69 he suffered first stroke loosing his speech ability. Nine years later he suffered another stroke and died in his sleep at the age of 76. His devoted wife, Sheila, took special care of him in his years of illness, describing their lives, especially Sir John's struggle after loosing his speech in the book "The Man who Lost His Language".</td></tr>
<tr><th colspan="2">Essential Nuggets</th><th colspan="2">Acceptable Nuggets</th></tr>
<tr><td>Born in 1923</td><td>V</td><td>Cause of death – stroke</td><td>V</td></tr>
<tr><td>Died at 1999</td><td>V</td><td>Lost his speech after a first stroke</td><td>X</td></tr>
<tr><td>Brilliant Historian</td><td>V</td><td>Wife named Sheila took special care of him in his illness</td><td>X</td></tr>
<tr><td>Chairman of the national Gallery 1974-1980</td><td>X</td><td>Author of "The Civilization of Europe in the Renaissance"</td><td>V</td></tr>
<tr><td></td><td></td><td>Renaissance Historian</td><td>V</td></tr>
<tr><td></td><td></td><td>Died in his sleep</td><td>V</td></tr>
<tr><td></td><td></td><td>His wife wrote a book about him and his illness – "The Man Who lost His Language".</td><td>X</td></tr>
</table>

Table 3.4 – "Who is Sir John Hale?" Answer and assessors lists. Check marks mark what nuggets were returned by the system.

Given a definition target, the assessor creates a list of facts acceptable as part of the answer. A subset of this list serves as a mandatory list of essential nuggets. Given a system response to a definition question, the assessor maps the nuggets of the system's response to the nuggets in his lists, calculating *precision* and *recall*. The mapping is based on conceptual-semantic similarity and independent of the syntactic features of the nuggets. *Recall* is computed in respect to the list of essential nuggets so the acceptable nuggets only serve as "don't care" in respect to recall. Since it is important to punish systems for providing non-relevant or too long answers (or answer nuggets) the *precision* is computed in respect to the general list of acceptable nuggets (including the list of essentials). Borrowing from evaluation of summarization systems [16;18;19], the length of the nuggets is a base to a crude approximation of the precision. The length based measure captures the intuition that users prefer the shorter of two definitions that contain the same concepts. Defining a length allowance *δ* to be the number of non white space characters for each correct nugget allowed, precision score is set to be 1 if a retrieved nugget is no longer than *δ*, otherwise the precision on a nugget *n* is is:

$\Pr ecision(n) = 1 - \frac{d(n) - \delta}{d(n)}$ , where d(n) is the number of non white-space character in *n*.

Let $L_Q^i$ be the list of nuggets retrieved by system $i$ for a question *Q*;

r be the number of vital nuggets $L_Q^i$ ;

a be the number of acceptable (non vital) nuggets in $L_Q^i$ ;

R total number of vital nuggets in the assessors list;

*δ* length allowance for a single acceptable nugget of information;

len be the number of non-white-space characters in an answer string summed over all answer strings in the response;

*β* be the precision and recall mixing parameter.

Then

$$\textit{(Nuggets Recall) NR} = \frac{r}{R},$$

$$\textit{Allowance} = \delta * (r + a),$$

$$\textit{Precision} = 1 - \frac{d(n) - \delta}{d(n)},$$

$$\textit{(Nuggets Precision) NP} = \begin{pmatrix} 1 & len < allowance \\ \sum_{nuggets\_retrieved} \Pr ecision(n) & otherwise \end{pmatrix},$$

and finally $F_\beta(L_Q^i) = \frac{(\beta^2 + 1) NP \cdot NR}{\beta^2 NP + NR}$

Table 3.5 Computation of F-score for definitional questions

Going back to the example of "Sir John Hale", a perfect answer and the two lists are presented in table 3.4. Looking at table 3.4, where *β*=5 (stating recall is 5 times important than precision) and *δ*=100 (stating the length of natural language nuggets is

100), $NR = \frac{3}{4}$ , NP = 1 (since the allowance is 100*(4+7)=1100 much longer than *len*). The F score for this toy example is 0.7572. Note that even though snippet #1 contains misleading data (John Hale didn't write short stories, the stories are attributed to another Hale), the system was not punished for it because of the big *allowance* and the big $\beta$ stating precision is not very important in this parameter setting.

This metric ignores the aspect of coherence and organization and combines only precision and recall. Section 2.4 described my goals in this work; one of them was building a QA system trying to achieve coherence of the retrieved answer. This chapter presents a baseline implementation of such a system; yet, this system requires a different metric in order to evaluate an answer with respect to its coherence along with its precision and recall. An alternative metric I developed – the FO (F score and Organization), taking coherence and organization into account is presented in appendix B[15].

In the next section I present the official TREC results followed by results of a tuned version of the baseline.

### 3.4 Results and Analysis

This section presents and analyses the TREC submitted results followed by the results of a differently tuned version of the baseline. The following section discusses the results, pointing on problems revealed in the baseline and suggesting a noble method to solve some of the problems. Remember that although the baseline is far from perfection, even the first, not-tuned, version did score very well and was ranked rather high by TREC assessors.

[15] The F score for definitional QA is problematic enough and not always stable. The alternative metric is even more problematic since coherence is a very vague concept and hard to evaluate in a consistent way. The alternative FO metric should be tested on larger scale before being used to evaluate systems and for that reason it is presented in an appendix and not as part of this chapter.

The average F-score of our TREC submission was 0.315, much better than the median (0.192) but still a bit behind the best system with the score of 0.55. These results place the system as the seventh out of 54 submissions. This average F score was computed over all the 50 questions in the TREC set but the system was scored null for 20 questions. The average F score over the 30 answered questions is 0.527.

Since the system was more focused on “who is” questions, I would also like to mention the distribution of our system F score over the different question types:
***Who is questions***: average F score: 0.392, improvement of 25 % of the overall score. The F score for “what is <an organization>” and for “what is <a concept> are: 0.268 and 0.15 in respect.

After a tuning session (independent of the TREC questions) I managed to improve the precision and the recall, boosting the overall system’s F score. The tuning session included tuning of two parameters 1) resizing the features subsets and 2) filtering threshold tuning.

| Query | Google Snippet |
|---|---|
| 1. William Shakespeare wrote graduated | of nine years between the time Shakespeare graduated from school ... It is believed that Shakespeare wrote The Comedy of ... **William Shakespeare died on April 23, 1616** |
| 2. William Shakespeare wrote graduated | became mistaken for the author known as "William Shakespeare."; ... exciting as the plays he wrote: his victories ... Paul Streitz graduated from Hamilton College, was |
| 3. William Shakespeare wrote graduated | The man who wrote Hamlet. ... **Hamlet has often been called Shakespeare's most autobiographical play**. ... story bears no relevance to the life of William Shaksper of |
| 4. William Shakespeare born wrote graduated won | fly_girl gr 8 The Life of William Shakespeare England's most ... first and best Italian poet and wrote mainly on ... By: Chris Slate David Herbert Lawrence Born on the |
| 5. William Shakespeare born wrote graduated won | James Joyce was born in Dublin and wrote all of his ... He lived and wrote in Paris,Rome,Trieste and ... Many critics consider William Shakespeare his only rival as a |
| **6. William Shakespeare born wrote** | According to some numerologists, **Shakespeare wrote The King James Version of the Bible at the age of 46**. ... **William Shakespeare was born in Stratford-upon-Avon** |
| **7. William Shakespeare born wrote** | **Shakespeare William Shakespeare was born in 1564** ... of Shakespeare comedies Shakespeare wrote many different ... times people say that William Shakespeare was and |
| **8. William Shakespeare born wrote** | whose twin brother died in boyhood), born in 1585 ... of Errors, **but in 1596, Shakespeare wrote Romeo and** ... Shop for William Shakespeare books at your local bookstore |

Table 3.6 – Google Snippets retrieved by various subsets of the FV

The effect of the resizing was a better quality of the retrieved snippets (see table 3.6 and appendix C).

The big subsets sometimes return irrelevant documents or corrupted snippets since Google tried to match the whole query. The longer the query is the bigger the chance to get corrupted snippets.  Long documents containing many of the required features will be represented by shattered snippets, sometime mixing sentences regarding other people (see rows 4 and 5 in table 3.6). Since the system is based on Google snippets and not on retrieved documents as a whole, the noise in the snippet is a function of the feature subset size, the length of the document and the distance between the terms of the query as they appear in the document. Resizing the FV subsets improves recall since the returned snippets are more accurate containing more focused details (see rows 6-8 in table 3.6).

The other tuned parameter is the filtering threshold. A bigger filtering parameter filters more documents, removes redundancies and improves precision. In the tuned run the filtering parameter was set to 85 instead of 70 in the submitted run.

The average length of an answer in the TREC submission was 2815.02. The average length of an answer in the tuned version was a bit more than half of the basic run – only 1481.62, causing a major improvement in the precision score. The improvement of the precision score is not necessarily linear to the change in the average length of an answer since the precision function is not linear (see precision function at table 3.5). In case the length of an answer drops under the allowance the precision is set to 1 as noticed in the "John Hale" example (section 3.3 and tables 3.4 and 3.5).

Looking at the results for question 1907 ("Who is Alberto Tomba?") in the TREC question set, the TREC submitted run returned 56(!) snippets of whom only 1 was counted as relevant by the TREC assessors (ignoring semantically close snippets). The tuned version returned only 16 snippets of whom 6 are counted as relevant improving the recall from 0 to 0.75 and precision from 0.014 to 0.37.  Examples of some differences between the TREC submitted answers and the tuned version answers could be found in appendix C.

After tuning the system achieved an F-score of 0.688 on the very same question set used in TREC. Further more, instead of 20 unanswered questions in the TREC submission[16], only 6 unanswered questions remained after the tuning (what helped in boosting the F score of the tuned version).

Since the runs sent to the TREC scored 0 for twenty (!) questions out of 50[17] (meaning that the overall F score is quite impressive), it might be interesting to look at the performance of the system in respect to the answered questions only. Calculating our TREC average for the non-zero F score questions shows an average F-score of 0.527. However, after further tuning we managed to reduce the number of unanswered questions to six only, obtaining an average F-score of 0.688.

## *3.5 Discussion*

According the official TREC results the hypothesis drawn in the beginning of the chapter is proven correct. This simple and rather naïve baseline achieved quite high ranking - 7th out of 54 submissions, though it is still far from perfection. Furthermore, the system was not evaluated in terms of organization and coherence at all. Looking on the system output (see appendix C) it is clear that the system would have scored badly on coherence if measured. In this section I discuss the results, analyze the weak points, and suggest more improvements following the rationale of the hypothesis (see section 3.1).

Since the main focus of our work was “who is” questions, I would like to discuss some issues noticeable in the results of the “who is” questions in a more specific way.

Past experience of using a dictionary such as WordNet shows quite good results [7; 17]. I decided to try and adopt this approach for “who is” questions and get as much information as possible from online encyclopedia such as biography.com. However, only less than half of the “who is” questions had entries in biography.com, the rest of the questions were answered using the Google snippets that were returned by searching predefined subsets of the features vector.

[16] The TREC assessors scored 20 questions with 0, some of them since no answer was retrieved while for some others – no relevant snippet was retrieved although the system returned an answer.

[17] See previous footnote.

Using the combination of a biographies-collection and the features-vector, we face some notable problems, mainly of two kinds.

The first problem of our system was coping with literature/fiction figures like “2130. Who is Ben Hur?” or “2322. Who is Absalom?” (see appendix C). Those figures have no entry in the external knowledge source since they do not have a real biography. When no biography could be found in the external source, the system is programmed to look for combinations of features that are connected to the lives of the people, thus the system will not find too much info related to fictional figures. We note however, that our tuning helped in improving the precision of this kind of questions, but had no real effect on the recall.

The other problem is of the opposite nature. For some figures, our system returned too much of irrelevant data, without the ability to filter the junk. Two questions of this type are “2082. Who is Anthony Blunt?” and “2125. Who is Charles Lindberg?”. There are too many people named Charles Lindberg and many institutes, schools and programs named after him. Most of the results seemed relevant but a second look reveals that not only it is not vital information; some of it is totally irrelevant.

Analysis of the distribution of the F scores over question types and over retrieval sources shows the importance of “formal” (or clean) collections in all aspects of evaluation. A “formal” collection is an *external knowledge source* such as dictionary, glossary , encyclopedia or biography collection. Answers retrieved from those kinds of sources were scored much higher than others.  Analyzing the good answers and understanding of the bad answers retrieved from Goolge, implies that using learning methods in order to learn the features and the optimal subsets of features will result in great improvement in the retrieved answers. Capturing the strength' and weakness' of the baseline, I think that using multi-document summarization methods on a small number of high quality sources (i.e. biography-like page) is better than collecting nuggets from very many sources. All the above suggest that identification of the high quality sources is important for good definition answering. In the next chapters I try to employ ML techniques for *text categorization* in order to identify the *good sources*. Text categorization prevents the system from getting lost in piles of redundant data.

In the next chapter (4) I review the basics of *ML*-based *text categorization* such as text representation, abstraction, feature selection and dimension reduction. I briefly present two chosen classification methods, and explain why those learners were chosen. Chapter 5 and 6 present the classification results of a naïve classifier (RIPPER) and the SVM classifier (in respect), and chapter 7 tests the integrated system on real definitional QA task.

# 4. Text Categorization and Machine learning

## 4.1 Introduction

The previous chapter presented the baseline system submitted to TREC 2003. Although the system was highly ranked by TREC the system is far from perfection. The definitional QA system sent to TREC 2003 was based on special *external knowledge source* and on *web retrieval*. The system exploits Google search engine, sending various queries then filters the returned snippets to form an answer from the remaining snippets (see section 3.2). Analysis of the results shows that Google snippets are not always sufficient. The baseline method is heuristic-dependent and since the system (just like the user) doesn't "know" the individual key terms related to each definition target, some important information nuggets may not be retrieved while some junk snicks in through the nuggets-filtering component, resulting in low precision and recall. Moreover, even when Google snippets contain all the relevant information and only the relevant information, the answer is poorly structured since combining many nuggets from different sources to form a coherent answer is a hard task, let alone the fact that each individual nugget, being a Google snippet, is not always well structured (see Appendix C for examples of answers).

| **Source of Answer** | **Average F-score** |
|---|---|
| Web (Google) | 0.304 |
| Glossary (WordNet) | 0.504 |
| Biography Collection (Biography.com) | 0.586 |

Table 4.1. TREC submission average F score according to answer source.

The hypothesis lying behind the baseline was that using *external knowledge source* or identification of *external knowledge source*-like documents, improves definitional QA systems. Performance analysis of the baseline reveals the hypothesis holds true. The analysis proposed is that retrieving the answer from a single source or very few good and reliable sources improves performance greatly (see table 4.1).

Unfortunately, most of the answers cannot be retrieved from external knowledge sources due to the collection's partial coverage. In those cases the answer should be retrieved from the web (or any other open corpus). Treating the web as a noisy *external knowledge source*, promotes the idea of filtering the noise and retrieving *external knowledge source*-like documents. Filtering of documents by genre can be done by *text categorization* methods. In the rest of the work I test the use of *Machine Learning* techniques for *text categorization* in order to achieve a better filtering.

For the remainder of the thesis we zoom in on biography-like documents categorization, used to improve answer for "who is" questions. The decision to concentrate on "who is" questions is explained in detail in section 3.2.3, however, a brief reminder of the main motivation is appropriate:

1. Search engines logs show that "who is" questions are more frequent than the two other types of definitional questions [26].
2. Biographies are easily treated as lists of facts; therefore categorization of biographies is very intuitive, therefore easier to model than a definition of organization or a concept.

In this chapter I review text categorization from a ML perspective, including some formal notations. I present the features of the two algorithms I use (Naïve classifier and SVM) and explain a number of decisions made, regarding text representation. In the next two chapters the performances of algorithms are analyzed and discussed. Chapter 7 analyzes the integrated QA system and the contribution of the learning biography-filter component to the QA system.

## 4.2 Text Categorization – Definition and General Introduction

Current *Text Categorization* (*TC* a.k.a. text classification, topic spotting or genre detection) lies in the crossroads of IR, ML and KE (Knowledge Engineering). In the past decade TC gained much interest in the information systems research community due a growing interest in, and a need for, applications and due to the availability of powerful hardware. Among the fields applying TC, one can find document indexing based on controlled vocabulary, document filtering, automated metadata generation, word sense disambiguation, population of hierarchical catalogues of the web and many more [22]. TC doesn't seem to have been used as a main ingredient in QA systems so far, although that many of the components that QA systems use, such as question classification, documents retrieval and named entity recognition do use it to some extent.

The goal of TC is to classify the topic or theme of a document, for example categorization of the Reuters text collection into categories like “mergers and acquisitions”, “earning reports” etc [Hayes et al. 1990]. *Text Categorization* is the task of assigning a Boolean value to each pair $\langle d_j, c_i \rangle \in D \times C$, where *D* is a domain of documents and $C = \{c_1, ..., c_{|C|}\}$ is a predefined set of categories[18]. A value of T assigned to each pair $\langle d_j, c_i \rangle$ indicates the decision to file $d_j$ under $c_i$, while value of F indicates a decision not to file $d_j$ under $c_i$.

More formally:

> The categorization task is to approximate the unknown *target function* $\Phi^* : D \times C \to \{T, F\}$ by means of a function $\Phi : D \times C \to \{T, F\}$ called the classifier, such that $\Phi$ and $\Phi^*$ coincide as much as possible.

The binary classification function is the standard way to look at TC. This is called *hard* categorization. For the purpose of this work a different kind of classification could be more appropriate. The QA system receives a stream of candidate documents and should filter/choose the best documents to get the answers from. For this purpose the classifier output should be a rank of a document according to its estimated appropriateness to a desired category. This process is called *ranking categorization*.

## 4.3 Machine Learning and Text Categorization

The baseline implementation of the QA system, presented in chapter 3, used a very simple and naïve categorization method somewhat close to the old TC technique known as Knowledge Engineering (*KE*). KE was popular in the 1980s. An example of this approach is the *Construe* system [Hayes et al. 1990] built by a Carnegie Mellon

[18] Note that is is also possible to do text categorization without knowing the categories in advance. This is done by using clustering techniques and the like, however, this kind of classifiers can only classify documents to categories without any “knowledge” about the meaning of a category. Obviously this approach is not suitable to the QA categorization task.

group for the Reuter agency. *Construe* was based on a set of manually defined logic rules in the form of

If (DNF formula) then (category)

The drawback of this approach is known as the *knowledge acquisition bottleneck*. In order to build such a system a knowledge engineer must be employed along with a domain expert. When dealing with many categories or once the set of categories is updated, new rules should be written, with cooperation of the knowledge engineer and the domain expert. Applying ML techniques could solve this problem of *knowledge acquisition bottleneck*. Using ML we build a learner that will be able to make the categorization decision after some training without the help of a domain expert. In case the set of categories is updated, the learner should be trained again only on a training set including the new category.

In order to train and test a learner/classifier, one needs a training set and a test set, which are taken from an initial corpus of documents $\Omega = \{d_1, ..., d_{|\Omega|}\} \subset D$. A document $d_i$ is a *positive example* of the category $c_i$ if $\Phi^*(d_j, c_i) = T$ and is a *negative example* of $c_i$ if $\Phi^*(d_j, c_i) = F$.

In this thesis *C ={biography, non-biographical}* and the classifier should learn to distinguish between a biography (or biography-like) document and non-biographical document. Finally, on integrating the classification component into the QA system, the classifier should distinguish between *biographies of John Doe* and other documents, even biographies. Although it seems a different task, one can look at it as two different levels of categorization. First, determining whether a document *d* is a biography, then determining that the biography subject is John Doe (or vice versa). Indeed, the two classifiers tested were also trained for both tasks, identifying a biography and identifying a biography of a certain person among many other biographies (see chapters 5,6). It is noteworthy that testing the learners on both tasks is not a necessity since it is most likely that Google, that is used as the system web-search engine, will not retrieve biography-like pages of Mr. Smith as response to query for John Doe.

## 4.4 Is This TC Problem Linearly Separable?

Classification problems can be divided into linearly separable and non-linearly separable problems. A dichotomy is linearly separable if the space of documents can be distinguished via a linear surface (line or hyper-plane). Some learners can learn only linearly separable problems, some learners can learn non-linearly separable problems but without reaching optimal solution while some other learners (SVM) can learn non-linearly separable problems by pumping the dimension of the problem and finding a linear separator in the higher dimension.

In the scope of this work it doesn't really matter if the biography categorization is linearly separable or not, since absolute separability is not as important as relative classification.  Humans can tell whether a document is not a biography, humans can identify pure biography when they read one, however, there are documents that even humans will hesitate whether they should be classified as biographies even though the documents are noisy and contain many biographical details. For this specific definitional QA task, we can make use of *noisy* biographies, although they would be ranked lower than *clean* biographies. A *clean* biography is a document that was written as a biography. A *noisy* biography is a document that wasn't written as a biography but as an article or review about the figure in question (i.e. NYT article presenting someone). I assume that a *clean* biography can be extracted from a *noisy* biography.

Moreover, although, collection of documents on the web is assumed not linearly separable (in terms of the biographies class), I believe most biography-like pages have some characteristics (structural, words frequency etc.) that enables the classification to be sufficient. This means that although sometimes some classifiers can reach no optimal solution, they can still achieve relatively good classification, which is sufficient to improve the QA task.

## 4.5 Data Representation, Document Indexing and Dimensionality Reduction

The first step in solving an ML problem is to decide on the data representation. Different data representations affect the learning process and lead to a different final hypothesis. A TC problem is no different; the learning domain is textual and data representation should be decided before feeding the document to a *learner*. Although it is very intuitive to represent a document by a vector of its unique words, it makes the learning process heavy and inefficient. Those vectors can be of very high dimension (for long documents) and contain many variations of the same word, this problem is called *overfitting*: the phenomena by which a classifier is tuned also to the *contingent* characteristics of the training set rather then the *constitutive* characteristics of the categories. It is shown that *overfitting* damages the learner's performance [22]. It was also shown that many sophisticated TC algorithms cannot operate efficiently on high dimensional vectors [22]. In order to overcome these problems an *indexing* procedure that maps a document *d* into a smarter compact representation of the content needs to be applied. Borrowing common IR techniques, stopwords removal, stemming and weighted-indexing seems an appropriate representation. Unfortunately, although it was proven to be a good way of representation for classic IR, the dimension of the vector is still very big and can reach more than a thousand terms per document; therefore most TC algorithms require a dimension reduction. The one exception to this requirement is the SVM learner in which instead of dimension reduction, dimension pumping is applied (see section 4.7).

Optimal dimension reduction is a hard task and various methods of dimension reduction are reviewed in Sebastiani's extensive survey paper [22]. The dimension reduction method applied in this work is described in detail in chapter 5.

Chapter 5 goes into the details of the data representation chosen for the naïve algorithm, using weighted vectors of n-grams of various lengths instead of tokens or words. Chapter 6 explains the SVM data representation. This section continues with a meta-tagging technique that was used for a better data representation, prior to the data transformation unique to each learner.

### 4.5.1 Document Abstraction Using Meta Tags

Let us look at an example. The string "Oren was born in 1976" is in not a typical line in a biographical document, while the abstraction "<NAME> was born in <YEAR>" might be significant. Imagine that 'years' (strings of four digits) are relatively frequent in biographies, however, since, obviously, the years mentioned in biographies are unique per biography, it is not easy to identify this characteristic unless years are marked by a meta-tag. It is assumed that biographies have other common terms, like "year" or "name", that are not identified as such unless they are clustered together. A cluster is a group of words (terms/tokens) with a high degree of pair-wise semantic relatedness, so that the group (or their centroids, or a representative of them) may be used instead of the terms as dimensions [22]. For the purpose of this work a set of seven clusters (see table 4.2) was defined in order to exploit the semantic similarity of tokens instead of making those terms redundant and even harmful for the learning process. Each cluster was marked with a meta-tag and all instances of a cluster were replaced by the cluster's marker.

**<NAME>** - the name of the subject of the biography.
**<YEAR>** - a chunk of four digits surrounded by white space, probably a year.
**<D>** - sequence of number of digits other than four digits, can be part of a date, age etc.
**<CAP>** a capitalized word in the middle of a sentence that wasn't substituted by any other tag.
**<PN>** - a proper name that is not the subject of the biography. It substitutes any name out of a list of thousand names.
**<PLACE>** - denotes a name of a place, city or country out of a list of more than thousand places.
**<MONTH>** denotes one of the twelve months.
**<NOM>** - denotes a nominative.

Table 4.2 - Abstraction tags for the biography classifier.

Collecting the training and test collections, each document was tagged as a positive or a negative example. The positive examples were also tagged with the subject of the biography. This tag was later used in the collection abstraction in order to substitute the biographies subjects with the subject tag <NAME>.

<PN> and <PLACE> tags substitute all occurrences of personal names or places (towns, states and countries) appearing in lists provided by the UvA LIT group[19]. Those lists were mined from the web and were successfully used in the factoid QA system [10].

[19] University of Amsterdam, Language and Inference Technology group.

<NOM> substitutes nominatives, assuming that biographies are rich in nominatives. The <YEAR> tag simply substitutes any string of exactly four digits while the <D> tag substitutes any string of digits longer or shorter than four.
<CAP> tag is used to since names and places lists are limited and do not consists of all the possible names and places. <CAP> and <D> tags cover this lack of information in an artificial way. The abstraction is not complete but working on a big corpus, the number of times it occurs is significant enough to justify its use.

Here is an example of the abstract representation of a short text. The next few sentences were taken from the biography of the jazz singer Abbey Lincoln:

> Lincoln, Abbey (b. Anna Marie Wooldridge) 1930 -- Jazz singer, composer/arranger, movie actress; born in Chicago, Ill. While a teenager she sang at school and church functions and then toured locally with a dance band.

The abstract representation is:

> <NAME> , <NAME> ( b . <PN> <CAP> <CAP> ) <YEAR> - <CAP> singer , composer/arranger , movie actress ; born in <PLACE> <CAP> . While a teenager <NOM> sang at school and church functions and then toured locally with a dance band .

It is noticeable, that some of the clusters, such as <CAP> and <PLACE>, <CAP> and <PN> and others may overlap. Reading the biography, one can notice that Abby was born in Chicago Illinois, but the abstractor couldn't recognize the token "Ill." standing for the state of "Illinois", therefore it was marked <CAP> for capitalized (possibly meaningful) word but not as <PLACE>, the same thing happens with the name "Wooldridge" that is not very common. Instead of marking it with <PN> it is marked with <CAP>.

All procedures and transformations described later on in chapters 5 and 6 are preformed on abstract-meta-tagged documents.

## 4.6 Naïve Classifier

This section gives a brief overview of the naïve classifier used as baseline for the biography-filtering component. This learning procedure was originally designed only as a comparative base for more sophisticated learning methods.

The naïve classifier is based on the RIPPER algorithm presented by Cohen and Singer [8]. It is motivated by its relatively good results, by performing extremely well across a wide variety of categorization problems and most of all by its simplicity. It was tested on big collections of noisy data – just the same as used for the definitional QA task. The naïve classifier learns a set of rules; if the set is compact enough "it makes it easier for users to accept a learned classifier as being reasonable" [8] opposed to other very complex algorithms like the SVM to be presented in the next section. Rules set can also be very easily converted to queries for a Boolean search engine [8]. This algorithm is also appropriate where *ranking* classification is more suitable than *hard* classification [22].

Although the abstract algorithm is extremely simple and straightforward, some rules and complications were integrated in order to adjust it to the task of TC and to the specific task of biography filtering. Next, I present the algorithm along with details unique to my implementation for TC. For a thorough comparative discussion about the algorithm and its mathematical properties see Cohen and Singer [8].

The naïve classifier consists on two main stages. The first stage is the term-selection, equivalent to the RIPPER's first stage – building an initial rule set. The second stage is optimization.

*Stage 1: Building the rules and dimension reduction*. In this implementation the terms, heuristically chosen, serve as the literals in the *rules vector* (for details please refer to chapter 5 section 5.2)[20]. In this stage the learner statistically learns the terms/rules by performing term extraction and dimension reduction. Two lists of the *k* most frequent terms are generated, *TLP* (term-list-positive) containing the positive example set and *TLN* (term-list-negative) containing the negative examples set. The vector $\vec{w}$ is initialized to be $TLP \setminus TLP \cap TLN$ - the most frequent terms extracted from the positive set that are not top frequent in the negative set[21]. Cohen and Singer argue against preprocessing and feature selection since they want to achieve more generalization and pay for this with a larger training set. However, feature selection is

[20] Cohen and Singer used the document tokens as the literals while I use various n-grams to improve context dependency and rules vector compactness.

[21] Note that this section only explains the abstract algorithm. Detailed explanation about term selection can be found in chapter 5 (section 4.5. In a brief note, the terms used are various length n-grams.

very widely used and proven to improve results especially when only a small training corpus is available and the rule set is preferably kept small so as to keep things simple [8; 20; 22].

*Stage 2: Optimizing and weights assigning*. In order to optimize the effectiveness of each rule in the *rules vector*, a weight is assigned to each rule/term in the vector[22]. The weights are set according to a normalized frequency score. Next, a threshold should be decided by testing several values for $\theta$ on a validation set. Notice that in evaluating a learning system, it is safer to use parameter setting (or tuning mechanism) that has been developed for problem different from the one being used as a benchmark. The RIPPER algorithm efficiency was tested using different using parameters of other classification problems [8], yet, in this work the threshold was decided by using the learner output for ranking documents of a validation set.
Looking for hypothesis of the form:
$c(d) = \begin{pmatrix} biography & score(d) > \theta \\ non - biographical & score(d) \leq \theta \end{pmatrix}$, the threshold $\theta$ is set to minimize false-positive and false-negative errors on the validation set. This threshold can be further tuned to perform better for the specific QA task as described in chapter 5.

To summarize, this naïve learning method was chosen for its simplicity. It was modified to fit text categorization and specifically TC for definitional QA. As argued by Cohen and Singer [8] this is a strong classifier, though very simple to understand, implement, train and control. This implementation depends on a strong term-selection and data representation that should enable learning using low-dimensional vectors and small training corpus. All those characteristics promote this algorithm as a good comparative base to the totally different approach represented by SVM learners to be introduced in the next section.

[22] This stage differs a bit from the original RIPPER's second stage for some optimization (pruning-like) was already employed by the n-gram mining in the first stage. In the RIPPER algorithm no optimization takes place at the first stage.

## 4.7 Support Vector Machines  Classifier

Support Vector Machines (SVMs) are currently regarded as the best general text classifiers [14; 22]. Opposed to many other classifiers, SVMs are capable of learning classification even of non-linearly-separable classes. It is assumed that classes that are non-linearly separable in one dimension may be linearly separable in higher dimension. SVMs offer two important advantages for TC [14; 22]:

1. Term selection is often not needed, as SVMs tend to be fairly robust to overfitting and can scale up to considerable dimensionalities.
2. No human and machine effort in parameter tuning on a validation set is needed, as there a theoretically motivated "default" choice of parameter settings which has also been shown to provide best effectiveness.

The idea behind SVMs is to boost the dimension of the representation vectors and then to find the best line or hyper-plane from the widest set of parallel hyper-planes. This hyper-plane maximizes the distance between two elements in the set. The elements in the set are the *support vectors*. It is noteworthy that theoretically the classifier is determined by a very small number of examples defining the category frontier – the support vectors. Practically, finding the support vectors is not a trivial task. SVMs applications use many transformations and tricks in order to perform those computations on possibly infinite-dimensional vectors [14].

In this work I use the standard free version of SVM-light v.5 [Joachim 2002]. SVM-light is used with its default setting with linear kernel function and no kernel optimization tricks.

As was stressed above, SVMs are very strong classifiers, capable of classifying non-linearly separable classes. SVMs are robust and no term selection is needed. The data representation is very simple while the learning process is very complicated. The SVMs approach was chosen as a contrast to the simplicity of the naïve approach and due to the features mentioned above, making it a general classifier that might be used "as is" for improving answering other types of definitional questions.

# 5. Naïve Biography-Learner – Training and Results.

This chapter reviews the training process and the results of the naïve learner that was built in order to classify documents according to a biography and a non-biography dichotomy. The previous chapter discussed a number of issues concerned ML and the specific algorithms used in this work. The naive algorithm used is based on the RIPPER learner. This chapter dives into the details of the data representation, term extraction, feature selection, the training process, and finally the results are presented.

## *5.1 Training Set*

The corpus we used as our training set is a collection of about 350 biographies. Most of the biographies were randomly sampled from biography.com while the rest were collected from the WWW. 130 documents from the New York Times (NYT) 2000 collection were randomly selected as negative example set. The volumes of the positive and negative sets are equal. The biographies in the biography.com domain were not written by the same biographer and have no strict form, however, a small number of biographies from different sources were added in order to eliminate an underling bias if such a bias exists. The biographies in the biography.com domain are clean in the sense that all of them were written as biographies. Some other noisy biographies such as biography-like newspaper reviews were added to enable learning of some of the features of non-formal biographies as well[23]. A small number of different biographies of the same person were manually added in order to force style variations.

[23] Having those documents added, it is assumed training set not linearly separable, just as the "Internet" collection of biographies.

## 5.2 Training – Stage 1: Term Selection and Dimensionality Reduction

The naive algorithm produces a vector $\vec{w}$ of rules with weights and a threshold *θ*. Given a document *d*, represented by the vector $\vec{x}$, the classifier decides the classification by checking whether the document's score, given by  the inner product of the weights vector and the data vector is greater than some *θ* as follows:

$$c(d) = \begin{pmatrix} biography & score(d) > \theta \\ non-biographical & score(d) \le \theta \end{pmatrix}, \text{ where } score(d) = \frac{\vec{w} \cdot \vec{x}}{length(d)}.$$

The dimension of the vectors is decided after term extraction and dimensionality reduction.

It is assumed that certain blocks of information are more important than others. We refer to *n-grams* of words as blocks of information. By term selection we try to catch salient blocks typical to biographies. The string "Oren was born in 1976" is not typical to a biographical document while the abstraction "<NAME> was born in <YEAR>" might be significant  due to the *context* of each word in the block (see 4.5). Not only that the abstraction of the year and the name increase similarity with the date of birth of other people, we assume that the 5-gram "<NAME> was born in <YEAR>" characterizes biographical information. This assumption leads to select n-grams typical to IE instead of tokens or stems typical to IR. The n-grams stand for the contextual rules described by Cohen and Singer [8].

*N-grams* of varying length were mined from both corpora where *n* varies from 2 to 5. The most frequent n-grams for each *n* were identified, and the most frequent *n-grams* that were found unique to the positive set were selected as the terms of the rules-vector and the documents vectors (see Table 5.1).

| |
|---|
| born <MONTH> <D> , <YEAR> |
| born in <CAP> , <PLACE> |
| <NAME> <NAME> |
| <YEAR> -- <YEAR> |
| ( TV series ) <YEAR> |
| at the age of <D> |

Table 5.1 Examples of frequent n-grams extracted from the training corpus.

### 5.2.1 Validation Set

The validation set was collected the same way as the training set. It contained 60 biographies, 40 randomly sampled from the biography.com domain and other 10 "clean" biographies were collected from various online sources, 10 other documents were noisy biographies such as newspaper reviews. The other 40 documents were non-biographical documents randomly retrieved from the web.

## 5.3 Training – Stage 2: Optimization and Threshold.

After the n-gram mining, the system also calculated the importance of each n-gram as a factor of its frequency and its length, assuming that a longer n-gram is a better marker and should weigh more than a shorter n-gram, and that a frequent n-gram is more significant than less frequent one (remember, those n-grams are not top frequent in the negative example set). The weights assigned to each term (n-gram) in the weights vector were computed in the following way:

$\frac{n \bullet f_s}{c}$, where *n* is the number of tokens in the string (the *n* of *n-gram*), $f_s$ is ratio of the string frequency in the positive examples set and the negative examples set and *c* is the total number of documents in the training set. This simple formula takes two important factors into account: the length of the *n-gram* and its relative frequency. The longer the *n-gram* is, the more likely it is to be typical biography pattern (context) and not just a coincidence. In other words the longer the *n-gram* is, it is more likely to represent a biography, and therefore we want to boost its weight more than the weight update in case of a shorter pattern. The *c* factor, which seems a constant and can be ignored, serves in comparing the learning process as a function of the size of the training set.

The vector $\vec{w}$ is now used to rank the documents of the validation set *V* in order to set a threshold that minimizes the false-positive and the false-negative errors. Each document $d_j \in V$ in the validation set is represented by a vector $\vec{x}$, where $x_i$ counts the occurrences of $w_i$ in $d_j$. The score of the document is the normalized inner product of $\vec{x}$ and $\vec{w}$ given by the function $score(d) = \frac{\vec{w} \cdot \vec{x}}{length(d)}$.

In the validation stage some heuristic modifications applied by the algorithm. In case the name-tag is absent the document gets the score of zero although other parameters of the vector are present. This is very reasonable since we not only that we want to distinguish biographies from non-biographical documents, our goal is also to use the classifier to improve the QA system and a question is always about a specific person[24]. The score of each document was normalized by dividing it by its length. This normalization is needed in order to prevent very long documents to obtain a high score only because of their length. Another rule was added for very short documents. In case a document is too short, we use its length to the power of 2 as the normalization factor. Those rules have benefits at some cost. On one hand it does unjust by downgrading very short and concentrated biographies, while on the other hand it rightfully downgrades documents containing the name of the subject of a query with many tags that appear to be irrelevant. It is assumed that very short biographies are not frequent in respect to short documents that can be mistaken for biography[25].

[24] I did check the classifier as a seperator between biographies and non-biographical pages in general (see section 5.6).

[25] Example of a good biography we might loose is: "**Hale, John Rigby, Sir** (1923-1999) British historian; chairman of National Gallery 1974-1980; wrote "The Civilization of Europe in the Renaissance" 1993. "
As far as I know this is the best match for the definition-QA pilot question "who is Sir John Hale?".
An example of a deceptive document we expose by this rule is:
"The Peabody Art Collection A Treasure for Maryland
Artist: Giuseppe Ceracci (1751-1802)
Title: Alexander Hamilton (1757-1804)
Date: 18th century
Medium: Marble
Dimensions: Height, 25"
Accession number: MSA SC 4680-20-0077
Return to Peabody Art Collection home page".
The page only gives details regarding a statue of Alexander Hamilton. It was retrieved for question "2174. Who was Alexander Hamilton?" of the TREC-12. The only relevant string in the page is "Alexander Hamilton (1757-1804)", obviously this document will not serve our purpose.

## 5.4 Test Collection

Before integrating the biography-filter into the QA system, the classifier was tested on a small test collection of 47 documents. All of the 47 documents were retrieved from the web in order to reflect the TREC baseline Google retrieved pages. Some of the documents are biographies that were found in search for biographies, some other are non-biographical pages that were retrieved while looking for biographies, some documents are noisy or tricky and contain many biographical information along with irrelevant information. We also added three completely random newspaper's articles in order to imitate real junk. Most of the documents contained the names of Arnold Schwarzenneger, Este'e Lauder, Johny Cash, Elgar Hiss, Alberto Tomba, Alexander Hamilton and Abu Sayaf. The latter four appeared in the TREC 2003 QA track and the baseline system used for out TREC submission (see chapter 3) scored rather poorly for those names.

The distribution of the documents in the test collection is presented in table 5.2.

| | | |
|---|---|---|
| Number of documents: 47 | Number of biographies and biography like docs: 16 (34%) | |
| Number of noisy biographies: 3. | Number of very short biographies: 3 | |
| Number of clean, detailed biographies: 10 (21%). | | |
| Johny Cash: 1 bio | Arnold Schwarzenegger: 3 bios | Alexander Hamilton: 3 bios |
| Elgar Hiss: 2 bios | Alberto Tomba:1 bio | Abu Sayaf: no bio found. |

Table 5.2 Distribution of documents in the test collection.

## 5.5 Results

The hypothesis produced by the learner is of the form:

$c(d)=\begin{pmatrix} biography & score(d)>\theta \\ non-biographical & score(d)\leq\theta \end{pmatrix}$. This function is represented by the vector $\vec{w}$ determined at the first stage of the algorithm and by the threshold *θ*. At the end of the training the threshold *θ* was set to 0.01.

We had seven runs on the test collection: one run for each name (as the collection is tagged differently according to the subject of the query) and one general run in which all the occurrences of the six names were tagged with the name-tag in order to identify a clear cut between biography pages and non-biographical pages.

In the General run for no specific name, the classifier had only 5 classification mistakes meaning 89% of success. The error distribution is as follows: 2 documents were classified false-positive and 3 documents were classified false-negative.

| | Before Tuning | | After Tuning | |
|---|---|---|---|---|
| Name | False Negative | False Positive | False Negative | False Positive |
| Cash | 0 | 0 | 0 | 0 |
| Schwarzenegger | 0 | 3 | 1 | 0 |
| Hiss | 0 | 3 | 1 | 1 |
| Hamilton | 0 | 3 | 1 | 0 |
| Tomba | 0 | 0 | 0 | 0 |
| Abu Sayaf | 0 | 0 | 0 | 0 |

Table 5.3 - Distribution of the classification error before and after tuning

The results of using the system to classify biographies for a specific person are more interesting. The results are presented in table 5.3. There is no contradiction between the figures in the general task of distinguishing the biographies from the non-biographical documents (pure genre detection) and each individual task (detecting biographies of a specific person from other biographies). The difference in the false-positive/negative score can easily be explained by the small change of tagging and abstractions.

Looking at the two left hand side columns of table 5.3, a slight problem with the false-positive results for three of the figures (Schwarzenegger, Hiss and Hamilton) is noticeable. In all cases few very noisy biographies passed the threshold and were classified as biographies. This problem can be fixed in two ways. One way is to try and add additional factors and complications to the weights vector in a way that those documents will be filtered out. Yet, the straight forward way, is to adjust the threshold manually. Our interest lies in getting very few documents of high quality. Recall or false-negative errors are not an issue as long as the system manages to obtain and high precision (low number of false-positive errors) on the top ranked documents. After tuning the threshold to 0.016 only one false-positive error while the size of the class of biographies is optimal (see tables 5.4, 5.5)[26].

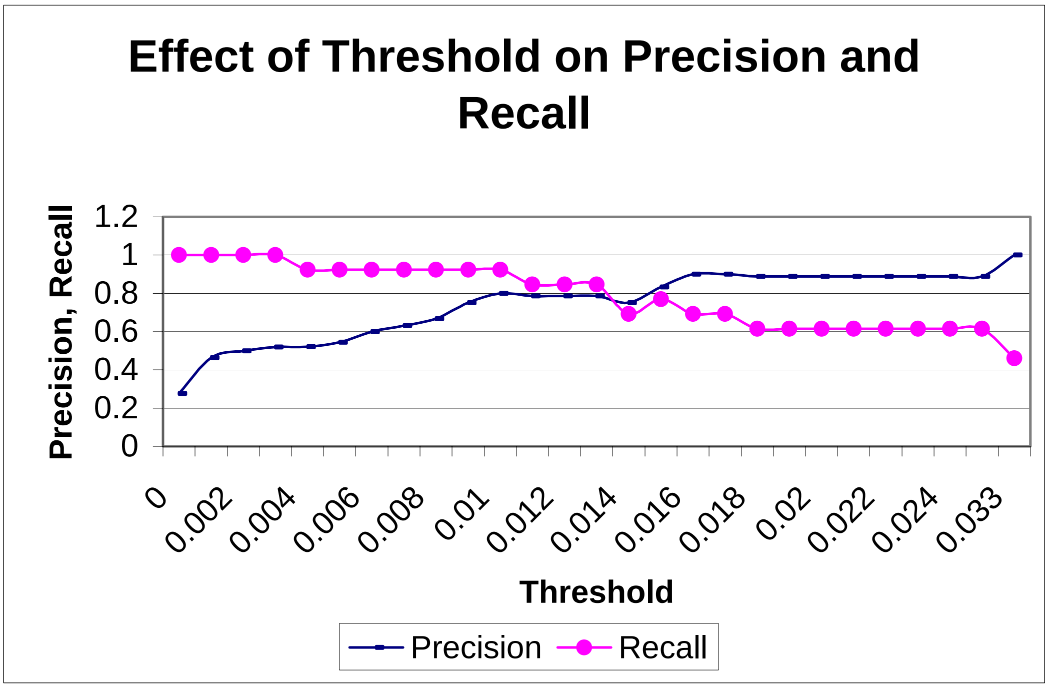

Table 5.4. Effect of threshold on Precision and Recall on in the general run

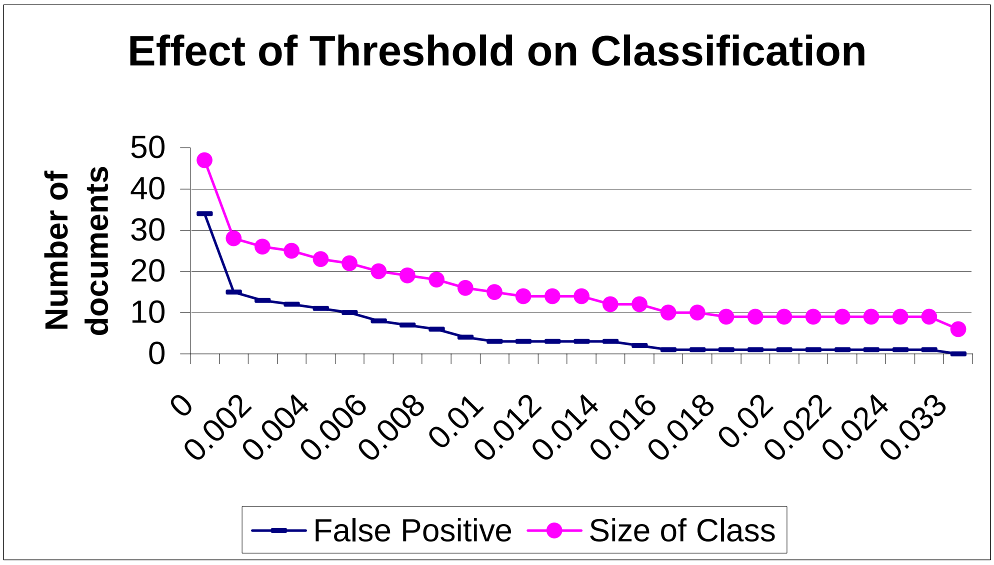

Table 5.5. Effect of threshold on False Positive error rate and on the size of the biography class in the general run.

[26] The one document that managed to snick in to our biographies section was a very detailed article about Elgar Hiss, the article contains many biographical details regarding Hiss and other important figures of his time. Those biographical details were boosted in the abstraction-tagging phase, which result in ranking this document at the top of the Hiss biographies list. It is clear that this is reasonable and within the borders of the standard deviation.

To summarize, the naïve classifier showed good results. The algorithm achieved its hypothesis using a relatively small training set and using low vector dimension. On the other hand, this classifier might be too task-specific and might require a different parameter setting in order to classify documents of other categories.

The next chapter presents another trainable classifier (SVM). Although the naïve classifier achieved quite good classification, the classifier used few rules that are very specific to biography classification. Indeed, this thesis takes "who is" question as a case study of the benefits of using text classifiers for definitional QA, but we hope to apply TC for other types of definitional questions as well. The algorithm presented in the next chapter is more general and no domain-specific rules are used in the learning stage. Chapter 7 presents an integrated QA system, evaluating the impact of both classifiers on a real definitional QA task.

# 6. SVM Biography-Learner – Training and Results

This chapter reviews the training process and the results of the SVM on the biography classification task. The implementation used is the SVM-light v.5 [Joachim 2002][27]. Again, two different tasks were tested. The first task was classifying documents according to the biography-non-biography dichotomy. The second task was distinguishing biographies of a certain figure from other, non-biographical documents regarding the same figure or biographies of other figures. Chapter 4 discussed few general issues concerning ML and some more specific details about the algorithms used in this work. Chapter 5 presented a naïve algorithm based on the RIPPER algorithm. The naïve algorithm achieved high precision and recall but its implementation includes specific rules for the specific task of classifying biographies. This chapter examines the SVM classifier. SVMs are much more complex than the naïve classifier but no special tuning is needed for different tasks. This chapter dives into the details of the data representation, the training process, and presents the results.

## *6.1 Training Set*

SVM-light was trained on the very same (meta-tagged) training corpus the naïve classifier was trained on. Since SVM is supposed to be robust and to fit big and noisy collections, no feature selection method was applied. The special feature lying behind the SVMs is dimension pumping; therefore the each document was represented by the vector of its stems. The dimension was pumped to include all the stems from the positive set, assuming the positive set, mostly extracted from biography.com and newspaper reviews, is rich enough to represent documents. Indeed the pumped vector dimension was 7935, the number of different stems in the collection. The number of support vectors discovered was 17, which is a way too small. Testing this model on the test set (the same test set used to test the naïve classifier, see section 5.5) yielded very poor results. It seemed the classification was totally random. Testing the

[27] Downloaded from http://svmlight.joachims.org/

classifier on smaller subsets of the training set (200, 300, 400 documents) showed signs of convergence, suggesting the training set is too sparse for the SVM.

In order to overcome the sparse data problem more documents were needed. The size of the training set was more than doubled. About 200 biographies and 250 NYT documents were added making the training set 9968 size. Just like the original training set, most of the biographies were randomly extracted from the biography.com domain while a few dozen biographies were manually extracted from various online sources in order to smooth bias of biography.com style if such a bias exists.[28]

### 6.1.1 Validation Set

Opposed to the naïve classifier where a validation set was needed in order to decide the threshold, no validation set is needed for the SVM training.

## 6.2 Training

The training was performed using the default configuration of the SVM-light. Linear kernel function was used and no kernel trick was applied. This time the highest feature index (vector dimension) was 7985 and 232 support vectors were found.

## 6.3 Test Collection

The test collection used was the very same test collection used to test the naïve classifier (see section 5.5). The collection contained 47 documents. The total number of biographies (and biography-like) documents was 16 (34%). 10 out of the 16 biographies were clean biographies, 3 other documents were noisy biographies and 3 more were very short and therefore less informative biographies. The 16 biographies deal with 6 different figures: Arnold Schwarzenegger, Johny Cash, Elgar Hiss, Alberto Tomba, Alexander Hamilton and Abu Sayaf (see Table 5.2 for exact distribution), the latter four names appeared in TREC 2003 and the submitted system (see chapter 3) scored poorly on those names. All other documents mention the names

[28] No such bias was noticed but maybe the learner could expose some underlying structure we couldn’t realize by ampling random biographies and reading them.

of the six figures and give some details concerning them, however, those document cannot be regarded as biographies, i.e. a newspaper report about Schwarzenegger's Californian campaign. This distribution of documents was specifically chosen in order to make the classification task harder and in order to enable the two different classification tasks – distinguishing biographies from non-biographical documents and distinguishing biographies of a specific person from other documents, possibly other biographies.

## *6.4 Results – Classifier Evaluation*

Just like the naïve classifier evaluation, seven runs were analyzed; one general run checked whether the classifier manages to distinguish between biographies and non-biographical documents and six more runs checking the ability to pick biographies of a specific person from a collection of many documents, some of them are biographies of different figures and some of them are non-biographical but concern with the figure in question.

### *6.4.1 General Run*

In the general run, 8 out of the 47 biographies were incorrectly classified (83% of success). The 8 errors consist of 2 false-positive errors and 6 false-negative errors. The 6 false-negative consists of 2 noisy biographies that were not identified as biographies due to their noise, 2 very short biographies that were not recognized as such since they are too short, therefore their vectorial representation was too sparse. The other two false-negative classified documents were clean-perfect biographies.

### *6.4.2 Specific-Name Runs*

In contrast to the fine results of the *general run*, the results for specific-name biography filtering tended to be an extreme failure. The SVM couldn't distinguish between biographies of two different persons. The classification was just the same as in the general run, meaning all the documents classified as biographies in the general run were wrongly classified as name-specific biographies. No differences in the classification were noticed in any of the 6 name-specific runs.

These results seem surprising at first glance, especially when compared to the naïve classifier that preformed relatively good for the name-specific tasks. Those results can be easily explained by the SVM document representation. As a document is represented by vector of dimension 7935, the name of the definition target contributes nothing to the classification, even where the name was substituted by a name tag (<NAME>).

## 6.5 Naïve Classifier Performance vs. SVM Performance

| | **General Task** | | | **Name-Specific Task** | | |
|---|---|---|---|---|---|---|
| | Success Rate | False Positive | False Negative | Success Rate | False Positive | False Negative |
| **Naïve Classifier** | 89% | 2.9% | 8.8% | 81% | 0% | 26.4 |
| **SVM Classifier** | 83% | 5.8% | 14.7% | N/A | N/A | N/A |

TABLE 6.1 Naïve classifier vs. SVM Classifier

Surprisingly, the naïve classifier outranked the SVM in both tasks. It is impossible to compare results of the name-specific task since the SVM was a major failure in that task. Comparing the general biography-categorization task the Naïve classifier achieved 89% of success while the SVM achieved 83% of success. These results suggest that the naïve classifier is more suitable for integration with the definitional QA system. This decision should be considered again in a wider context. In this work the interest in text classifiers doesn’t lie in pure text categorization but in improvement of the definitional QA system. The reader should keep in mind that the system gets the documents to categorize from Google after sending a query with the name of the definition target and possibly some other words to improve relevancy of the documents retrieved. Google itself applies filtering and ranking algorithms and outputs a ranked list of documents Google finds relevant. We assume it is not likely that a biography of X (or documents of whom X is the subject) will be retrieved for a query about Y (unless X and Y are correlated). The challenge Google-output presents is of identifying the most biography-like documents among the set of all retrieved documents. Relying on this assumption, the failure of SVM to distinguish biographies of different persons looses its relevancy since, under this assumption, no biographies of other figures will be retrieved.  This assumption means that although theoretically-wise, the task is the name-specific task, practically-wise the task is equivalent to the general task of distinguishing biographies from non-biographical documents. This being said, one should only compare the results of the general run in deciding which classifier to choose.

Moreover, since the QA biography filtering component should identify one or very few “good” (biography-like) documents that would serve as the basis to the final answer retrieved by Google, the system is indifferent to false negative errors as long as the false positive error rate is low and as long as some documents are being classified as biographies. The false-positive rate is equal for both classifiers.

Two other aspects should be taken into account before deciding which classifier to use: simplicity and portability. While the naïve classifier is very simple to understand but requires a different parameter setting (and maybe even slightly different implementation) for each classification task, the not intuitive SVM could be applied to any other classification task (according to TREC questions types) without any change. In case both classifiers present close performance in biography-categorization, the more portable classifier has an advantage.

Finally, the reader has to keep in mind that the purpose of this work was to serve as a proof of the concept that using ML techniques of TC improves performance of definitional QA systems. It is possible (and very reasonable) that further tuning can improve classification of both classifiers. This optimization is beyond the scope of this work.

The next chapter reviews the integrated QA system, presents the results and evaluates the contribution of the biography-filtering component to definitional QA system.

# 7. Integration of a Biography Classifier With the definitional QA system.

The hypothesis lying behind this research is that the use of a *text classifier component* improves performance of the definitional QA system. This hypothesis was promoted by analysis of the evaluation of number of different systems [7; 10] showing that answer nuggets that were collected from a small number of sources have better recall and precision, achieving higher score in general. This observation was also supported by analysis of the official TREC results. Another advantage gained by using small number of sources to generate an answer lies in the relatively small effort needed to present the answer in a coherent and organized way.

Previous chapters examined and compared two TC methods in lab conditions. This chapter examines the integrated system in "real world" definitional QA task.

## 7.1 Integrated Architecture

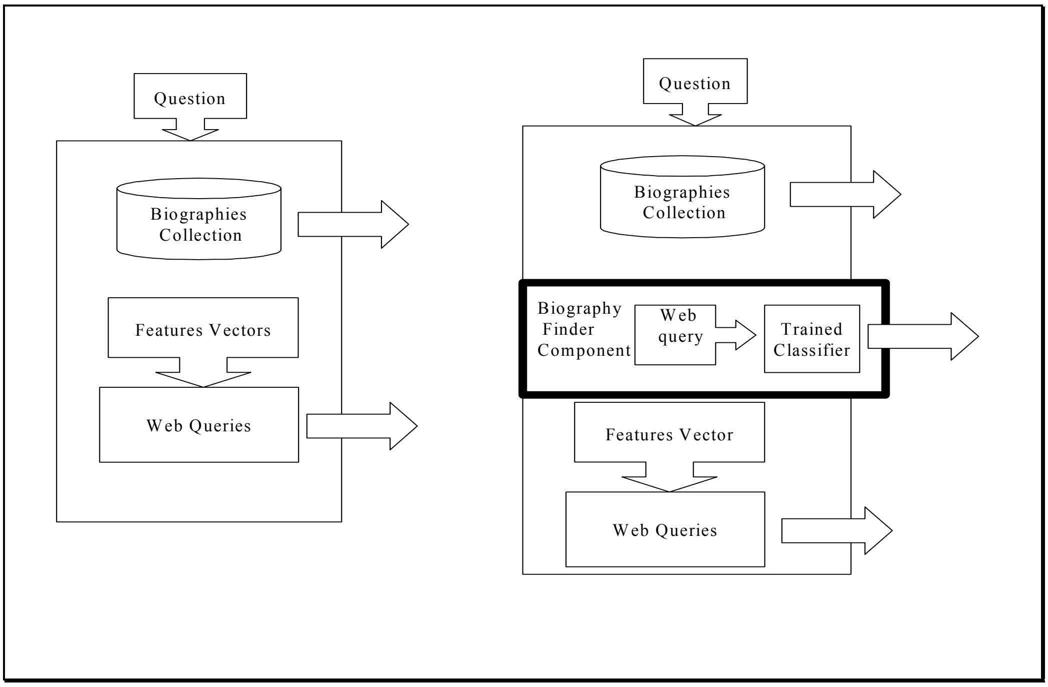

Table 7.1 Baseline Biographical Component architecture vs. New biographical Component

In order to test the hypothesis a new architecture was proposed. In the baseline architecture the biographical component operates in two steps:

1. Look for a biography of the definition target in “expert” biography collection. In case no answer found:
2. Formulates a series of Google queries and use the snippets as an answer.

The new system operates in the following manner:

1. Look for a biography of the definition target in “expert” biography collection. In case no answer found:
2. *Biography finder*: Formulate few simple Google queries, extract top *n* pages of each query, classify the pages as biographies and non-biographical documents and return top ranked biography-like document. In case no biography-like document found:
3. Formulates a series of Google queries and use the snippets as an answer.

The two Google queries sent by the biography finder were “<definition target> biography” and “definition target”. The system processed the top 20 answers for each query.

## 7.2 Test Set

This work is focused on using TC to improve results for “who is” questions (see section 4.1). “Who is” questions occupied 60% of the TREC definitional questions set. The system’s baseline average F-score in TREC 2003 was 0.316. The average F-score and the median for “who is” questions was 0.425, while 11 questions, more than third of the “who is” questions were scored less than the system’s average (in fact 8 questions were scored null). Those 11 low-ranked questions were chosen as test set in order to see how the integration with the TC component improves results.

## 7.3 Results

In order to test the integrated system two runs were performed. In the first run the naïve classifier was integrated into the QA system and in the second run the SVM classifier was integrated. The biography-finder component sends two Google queries for each definition question and retrieves the top 20 documents returned by Google, meaning each question gets 40 documents from which the system should find a biography. The *biography finder* then classifies the documents to biographies and non-biographic documents. The distribution of documents that were classified as biographies can be found in table 7.2.

| **Question** | **Naïve Classifier** | **SVM Classifier** |
|---|---|---|
| 1907. Who is Alberto Tomba? | 2 | 4 |
| 2060. Who is Albert Ghiorso? | 8 | 2 |
| 2267. Who is Alexander Pope? | 13 | 11 |
| 2274. Who is Alice Rivlin? | 3 | 2 |
| 2322. Who is Absalom? | 2 | 3 |
| 2324. Who is Nostradamus? | 1 | 1 |
| 2332. Who is Machiavelli? | 3 | 6 |
| 2024. Who is Andrea Bocceli? | 1 | 1 |
| 2208. Who is Al Sharpton? | 2 | 6 |
| 2006. Who is Aga Khan? | 4 | 1 |
| 2130. Who is Ben Hur? | 2 | 1 |

Table 7.2 Number of Documents classified as biography by both classifiers (including the false-positive documents)

The QA system should return one document per question. However, in the classifying process, more than one document was classified as a biography (see table 7.2) and the system returned a ranked list of biographical ducuments. The evaluation of the system was held in respect to two aspects: the average F-score of the top biography in the lists, and the classification power of the system. The classification power is measured by the false positive ratio for all questions in the test set.

Since this work tries to test the contribution of *text classifiers* to QA, the evaluation of the integrated system was quite harsh. Where the biography finder returns a “biography”, the assessor first checks whether the document is a pure biography or not. In case the document is not a pure biography the F-score given to this question is zero. This F-score is misleading and should be actually much higher, since even the non-pure biographies contain many biographical pieces of information that should contribute to the answer F-score. The *nuggets precision* and *nuggets recall* were computed only for pure biographies. This harsh evaluation was used since the primary goal of this research was to imitate *external knowledge sources* by filtering the noise from the web. Achieving high F-score by other means is very nice but it is only a side effect.

| Alberto Tomba | Albert Ghiorso |
|---|---|
| Alberto Tomba Born: Dec. 19, 1966 Italian alpine skier winner of 5 Olympic medals (3 gold, 2 silver); became 1st alpine skier to win gold medals in 2 consecutive Winter Games when he won the slalom and giant slalom in 1988 then repeated in the GS in '92; also won silvers in slalom in 1992 and '94; won 1st overall World Cup championship along with slalom and giant slalom titles in 1995 | • Presently a member of the Nuclear Science Division.<br>• Staff Member, University of Chicago Metallurgical Laboratory, 1942-46. Senior Scientist, Lawrence Berkeley Laboratory, 1946-82.<br>• Senior Scientist Emeritus, ditto 1982-present.<br>• Education: B.S. Electrical Engineering, University of California, Berkeley, 1937; honorary Doctor of Science degree, Gustavus Adolphus College, 1966; American Chemical Society Award: Nuclear Applications in Chemistry, 1973<br>• Co-Discoverer of 13 elements, 95-106 and 110 (to be confirmed). |

Table 7.3 Pieces of information extracted from very noisy biography-like pages.

To demonstrate the previous point, table 7.3 presents fractions of a biography-like pages returned for different questions ("who is Alberto Tomba?" and "who is Albert Ghiorso?"). As one can see, the information is important and will achieve a reasonable *recall*. The *precision* in those cases is low, but remember that $\beta$=5, meaning recall is much more important than precision. The reason those questions were scored 0 (in this chapter) is that those pages are very noisy biographies containing information about other or many other commercials and irrelevant information. Again, this work aims to find relatively clean biographies therefore I treat those results as a failure, not calculating the F-score of very noisy documents.

### 7.3.1 Definitional QA System Integrated With Naïve Classifier

The figures of how many documents were classified as biographies can be found in table 7.2. The total number of documents that were classified as biographies is 41 (out of 440 retrieved documents). However, analysis of the results reveals that the false positive ratio is unexpectedly high; only 20 of the 41 chosen documents were biography-like pages. Looking at the biography-rank of each document, 4 questions out of the 11 were answered by the biography finder (meaning a clean biography was returned), all four questions were scored 0 (zero) at the TREC while now their average F-score is scored 0.732, improving the average F score by 19% to 0.375.

Looking at the subset of the 30 "who is" questions, the biography finder improved the average F-score by 9.6% to 0.4659.

Analyzing the questions in the test set, 2 of the 11 questions in the test set couldn't be answered by the biography-finder component since the definition target was a literary figure therefore no biography is available[29] (see section 3.4). 2 other questions had no biography in the 41 chosen documents (Agah Khan and Andrea Bocceli). The rest of the returned answers were very noisy biographies, therefore are treated as wrong answers although useful information could be extracted (see table 7.3).

Inspite the major improvement in the system performance, the statistics reveal two problems. First, the false positive ratio is disturbing. It seems that although the system managed to identify biography-like documents, it has high false-positive ratio and too many errors in filtering out some of the non-biographic documents. This happens when the documents retrieved by Google are very noisy, and cannot be regarded as biographies by human assessors although they contain many biographic details. Second, most of the definition-targets had biographies retrieved and even classified as biographies but the biographies were ranked below another non-biographical documents, therefore the biography was not presented as an answer[30]. Improving the system's accuracy by further tuning may improve the system much more.

[29] The two questions are:
2322. Who is Absalom?
2130. Who is Ben Hur?

[30] I.e. 2 out of the 40 documents retrieved for the question "who is Alberto tomba?" were classified as biographies. One of them was a real biography and was scored 0.01825 but the other document that described the films Alberto Aomba was acting in was scored 0.0185, outranking the real biography. This phenomena was observed in 3 other questions.

### *7.3.2 Definitional QA System Integrated with SVM Classifier*

The results of the integrated system using the SVM classifier do not radically differ from the naive classifier results, although the two classifiers identified different sets of documents as biographies (of course the two sets share a large intersection).

The numbers of documents classified as biographies can be found in table 7.2. The total number of documents that were classified as biographies is 38 (out of 440 retrieved documents). However, just like in the naïve classifier case, analysis of the results reveals that the false positive ratio is unexpectedly high; only 18 of the 38 chosen documents were biography-like pages.

The SVM classifier managed to return a clean biography to 5 questions. The average F score for those questions is 0.674 instead of 0, improving the average F score of the system by 21% to 0.383.

Looking on the subset of the 30 “who is” questions, the biography finder improved the average F-score by 9.7% to 0.4665.

No biographies were retrieved to 4 of the 11 definition targets of the test set – the same four definition targets the naïve classifier didn’t find biographies. The statistics reveals the same problems noticed using the naïve classifier – relatively high false-positive error ratio and bad ranking of the classified biographies

### 7.3.3 Naïve classifier vs. SVM classifier – Results Analysis

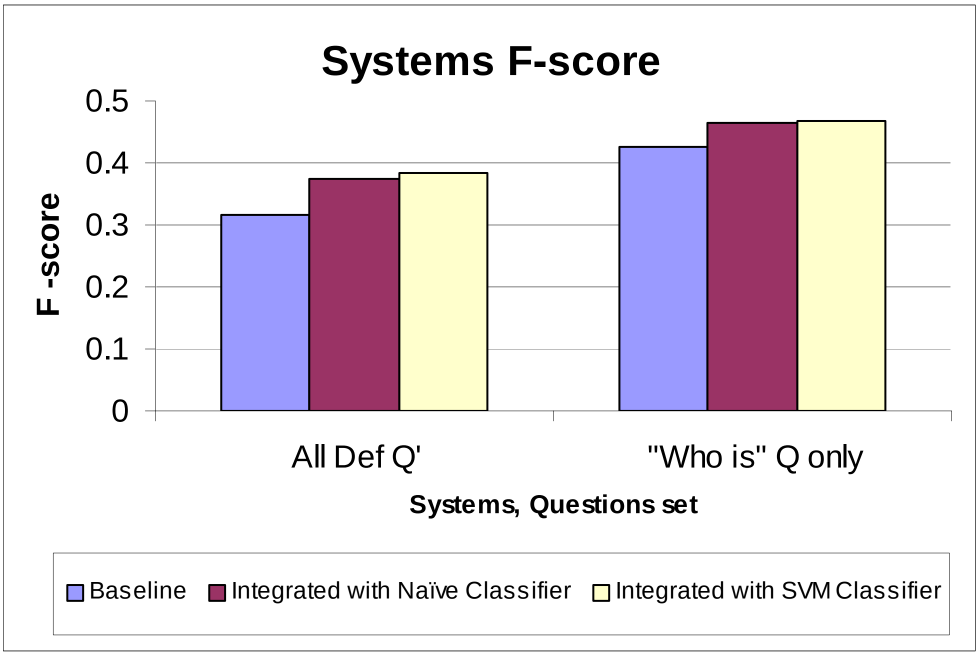

Table 7.4 systems F scores

The results of the experiments using both classifiers are quite similar. This thesis is meant to serve as a proof of concept to the hypothesis that using TC methods greatly improves the baseline QA system in an elegant way. Although improvement of almost 20% was noticed, this section is not celebrating the achievement but is focused on the problems exposed. The improvement rate is similar and the problems discovered are similar as well. However, the advantage of the naïve classifier is clear when trying to understand the problems. Using the naïve classifier, the errors snick in due to the special content of the documents retrieved. Documents containing lists of publications or dates get high rank due to their structure for the classifier identifies n-grams patterns that are typical to biographies. Further tuning or different patterns search (different feature selection), might solve both problems.

Understanding the problems behind the SVM is much more difficult. The SVM model consists of 7965 dimensions vectors and it is almost impossible to perceive the reasons something is going wrong. Opposed the naive classifier in which the classification method is transparent to the user/researcher/developer the SVM results analysis is based on intuition and each new hypothesis should be tested multiple times.

One reason that might explain results (F-score and false positive ratio) for both classifiers is the free structure of a web page and the building of the training set. The training set is based on a positive examples set and a negative examples set. The positive set was built from hundreds of biographies taken from an expert system with few web-mined biographies and few noisy biographies. The negative set is a collection of hundreds of New York Times articles, taken from NYT 2000 collection. Both sets are diverse in the sense that no common writing style cause a bias in the learning process, however, the collections are clean in the sense that no web-page designs effect the document representation. Trying the integrated system on real QA task and not only a toy classification task, showed a great improvement indeed but also exposed the fact that the training sets are too clean. Looking at some of the documents that were classified as biographies, the documents, although being cleaned of HTML tags, still contain page-design patterns like list of words of menus, headers or links titles[31]. All those patterns were absent from the training set and therefore might cause discrimination in the "real world" classifying process. Note that testing the classifier (sections 5.5 and 6.3) the test set consisted of web-retrieved documents but only of the main text of the documents, ignoring the designs. The training set was built this way since a biographies collection is the only source one can obtain hundreds of biographies without manually looking for them on the web and manually copying the page HTML code. Moreover, the clean biographies should give the learner a clear picture of what is a biography. The test set was built manually in order to build a deceiving test collection in order to make it difficult for the classifier to classify the documents correctly, however, collecting the documents for the test set, the web design was ignored in order to test and compare the classifiers performance in lab conditions in order to find which classifier suits this task best.

This bias should be fixed in one of the following ways; both methods are beyond the scope of this thesis:

1. Train the classifiers again in less sterile environment. This requires a great investment in collecting many hundreds of random biographies from the web.

[31] All strings that appeared between the HTML tags. The HTML tags were cleaned but the information was kept. Distinguishing the main text body its headers and titles from the site/web page designs is a different task, requires an extensive research.

This should be done manually (with the help of the classifiers presented above in order to filter some of the documents).

2. Pre-processing the retrieved documents in order to clean them from all kinds of web-design dependencies[32].

# 8. Conclusions

## 8.1 Summary

The naïve baseline approach to *definitional QA* was proven relatively effective. The baseline approach is based on IR methods, using several heuristics to formulate smart queries optimizing *recall* and using semantic distance metric in order to optimize *precision* (see chapter 3). The baseline was submitted to TREC 2003 and was ranked 7th among 54 submissions. Analyzing the questions which the baseline system didn't score well, analyzing the questions the system scored very well and analyzing previous work on definitional QA, this thesis promotes the idea that using *text categorization* methods can improve results in an elegant way. It was argued that not

[32] Note that some of this work should be done anyway in order to present the answer in the most coherent and organized way. This is the final task of definitional QA system but current research is still far from it. The last TREC, stating the current research agenda, gave coherence no importance at all.

only it improves *precision* and *recall*; it also improves the coherence and the organization of the answers.

As a test case, two totally different *machine-learning* algorithms for *text categorization* were tested then integrated into the QA system. One learner is IE oriented and although the features selected automatically, the implementation makes use of domain expert, the other learner is the robust SVM. The learners were trained to distinguish biographies from non-biographical documents, in order to test the hypothesis on a subset of the definitional questions - the set of "who is" questions which occupies the majority of definitional questions found in search engine logs.

## *8.2 Conclusions*

The integrated system was tested on a small toy-set, the set of questions on which the baseline scored poorly. Checking improvement only by "pure biographies", performance of the integrated system was improved by 9% over the whole range of definitional questions and was improved by 10% over the set of "who is" questions. This evaluation of improvement was measured using the F score measure, in respect to precision and recall alone, ignoring the coherence and organization of the answer. It was also argued that using TC component also improves the coherence and organization of the answers but measuring coherence is not straight forward as the F-score measure is (see appendix B).

Inspite of the good results, the system is still far from being perfect and could be improved in many levels –learning optimization, improving coherence, improving training etc.

Consequently, tested only on a toy test set, this thesis serves as a proof of concept that integrating a *trainable text categorization component* is a noble way to improve performance in many levels, suggesting that a further research should be done in order to fully use the advantages of text categorization in the aid of definitional QA.

## 8.3 Future Work

As stressed more than once, this thesis was an initiative work trying to test the applicability of text classifiers in the definitional QA research. This thesis is only a proof of concept and much work should still be done. Future research should concentrate in few different levels:

1. Optimizing the learning process, finding size of training sets on which the learning process converges.
2. The integrated system was tested only on a small toy-set of questions. Experiments should be performed on bigger sets to get sounder results.
3. Testing other types or variations of learners that might be applicable to definitional QA systems.
4. Elaborating the use of text classifier to types of definitional questions, other than "who is" questions.
5. Further explorations of automatic summarization techniques in order to improve retrieved documents quality and improve coherence and organization of answers.

# Appendix A – Glossary

***Closed Corpus*** – the collection of which we mine from is well defined, static and known.

***Definitional Questions*** – questions that seek a set of interesting and salient information items about a person, an organization or a thing (What is a battery? Who was Alexander Hamilton? What is Yahoo?).

***Domain Independent/ Open Domain*** – the system is not restricted to deal with a specific field.

***Domain Specific/Close Domain*** – the system can only handle queries regarding a specific field.

***External Knowledge Source*** – collection of structured of semi-structured documents, usually organized around a particular topic. It is *external*, because it has been created by others and is accessible either via the Internet (i.e. biography.com) or by other means (locally installed WordNet).

***Factoid questions*** – questions that seek short, fact-based answers (Where is the Taj Mahal? Who is the president of the U.S.?).

***Open Corpus*** – the collection to mine from is dynamic.

***Text Categorization*** (text classification, topic detection, genre detection) – Where $D$ is a domain of documents and $C = \{c_1, ..., c_{|C|}\}$ is a predefined set of categories, the categorization task is to approximate the unknown *target function* $\Phi^* : D \times C \to \{T, F\}$ by means of functions $\Phi : D \times C \to \{T, F\}$ called the classifier, such that $\Phi$ and $\Phi^*$ coincide as much as possible.

# Appendix B – The FO Score: A Unified Metric for Precision, Recall and Organization

Chapter 3.3 describes in detail the F-score metric used by TREC assessors for definitional QA systems evaluation. In this thesis, I adopted the F-score metric in order to compare TREC results with other definitional QA systems I implemented, yet, definitional QA systems require a different metric in order to evaluate an answer with respect to its coherence along with its precision and recall. F-score is not the optimal measure for definitional questions for two reasons: On one hand the F-score is not strict and different assessors might build different lists of essential and acceptable nuggets. Even the same assessor might consider a nugget as essential at one run and as acceptable on another run [10; 27]. On the other hand, the F-score is totally ignorant of the coherence and organization of the answers. Looking at the examples in Appendix C, it is clear that coherence should be taken into account in the future.

While working on this thesis, one of my goals was to find a noble method that achieves high F-score along with good coherence and organization. In order to measure organization I developed a unified metric – the FO metric (FO standing for F-score and Organization). This appendix presents the FO metric, although I didn't use this metric in the evaluation of the systems presented in this work. There are two reasons for not yet using the new metric. Firstly, I'd like to compare my results to a standard and well-accepted metric such as the F score used in TREC. Secondly, although the FO metric is aimed to represent organization in a quantitive way, the FO metric is not stable and still leaves a lot of freedom to the assessor to rank an answer as he finds appropriate.

A definitional QA assessment should be based on five factors, quite close to the factors used by the TREC assessors:

1. Number of retrieved snippets.
2. Number of relevant snippets.

3. Recall, computed by the *NR* with some freedom given to the assessor (see section 3.3 and Table 3.5).
4. Precision, computed just like the *NP* (see section 3.3 and Table 3.5).
5. Organization (0-1: 0 – the answer is not organized at all, 1- the answer can be presented as a perfect answer given by an intelligent human agent)[33].

The score for organization is affected by few factors: the answer consists of a number of disconnected nuggets, the junk inside each single nugget, the organization of each single nugget etc. In order to determine the importance of the organization I use an organization factor γ, varies between 0 and 1. The higher $\gamma$ is, the organization score has a stronger effect on the overall score. When $\gamma$=0 the overall score is the simple F-score presented in section 3.3, while $\gamma$=1 denotes that the final score is totally controlled by the organization score.

The FO  score is computed as follows (see Table 3.5 for variables definitions):

$$(3) \qquad FO_{\beta,\gamma}(L_Q^i) = ([1-\gamma] + O(L_Q^i)) \cdot F_\beta(L_Q^i)\,,$$

which is actually equivalent to:

$$(4) \qquad FO_{\beta,\gamma}(L_Q^i) = ([1-\gamma] + O(L_Q^i)) \cdot \frac{(\beta^2+1)NP \cdot NR}{\beta^2 NP + NR}\ .$$

In order to use this metric in future experiments, more tests should be taken checking the stability of the organization score given to many assessors. If it turns out that the variance between assessors is small, the FO metric is sound to use.

[33] A score of 1 was given when the answer was retrieved from another expert system or when one perfect snippet was retrieved (or in the case of more than one perfect snippets was retrieved).

# Appendix C - Examples of TREC Submitted Answers vs. Baseline Tuned Version Answers

## *Summary*

This appendix presents 4 examples of answers to questions on which the tuned baseline differs from the TREC submitted run.

The 4 questions are:

1. **What is the golden parachute?** - There is no big difference between the two runs. Since both TREC snippets are conceptually close, the tuned version filters one of the TREC snippets out, due to the higher filtering threshold that is more sensitive to similarities.
2. **Who is Alice Rivlin?** – TREC submission finds many different snippets, some are relevant while the tuned version finds a single perfect snippet, boosting precision, recall and F score.
3. **Who is Alberto Tomba?** – Both runs are far from giving a good answer but while the TREC system is scored very poorly for precision, recall and F score, the tuned version filtered most of the junk improving precision and managed to obtain more informative snippets, improving recall therefore the F score is slightly better.
4. **Who is Ben Hur?** – The TREC submission returned few "mysterious" results not connected to anything. This happens since the TREC submission was tuned to search for relatively bid subsets of the FV (i.e. "born, graduated, achieved, suffered, won, book") and no relevant document could be found for this query. The tuned version trying smaller subsets achieved results.

## *Question 1905 (What is the golden parachute?)*

**TREC submission:**

1905 UAmsT03M1 1 NYT19980820.0014 A golden parachute is contractual agreement to assess, in advance, the amount of compensation payable to dismissed executive.

1905 UAmsT03M1 2 NYT19980820.0014 of the takeover. &quot;golden parachute&quot; is sweet deal for an executive who loses his or her job in takeover. The benefits might

**Tuned system answer:**
2 rel run0 A golden parachute is a contractual agreement to assess, in advance, the amount of compensation payable to a dismissed executive.

## *Question 2274 (Who is Alice Rivlin?):*

**TREC submission:**
2274 UAmsT03M1 1 NYT19990608.0356 (1957), was deputy assistant at the Department of Health, Education and Welfare (1966--69), and then became the first director of the Congressional Budget Office (1975--83)
2274 UAmsT03M1 2 NYT19990608.0356 C
2274 UAmsT03M1 3 NYT19990608.0356 D
2274 UAmsT03M1 4 NYT19990608.0356 Daughter of nuclear physicist, she graduated from Bryn Mawr (1952) and earned Ph
2274 UAmsT03M1 5 NYT19990608.0356 economy, she wrote several studies of economic problems, including Economic Choices (1987) and Caring for the Disabled Elderly: Who Will Pay? (1988)
2274 UAmsT03M1 6 NYT19990608.0356 from Radcliffe (1958)
2274 UAmsT03M1 7 NYT19990608.0356 In 1993 she was appointed deputy director of the Office of Management and Budget in the Clinton administration
2274 UAmsT03M1 8 NYT19990608.0356 respected analyst of the U
2274 UAmsT03M1 9 NYT19990608.0356 Rivlin, Alice (Mitchell) 1931 -- Economist, government official; born in Philadelphia
2274 UAmsT03M1 10 NYT19990608.0356 S
2274 UAmsT03M1 11 NYT19990608.0356 She became staff member of the Brookings Institution in Washington, D
2274 UAmsT03M1 12 NYT19990608.0356 She then became director of economic studies at the Brookings Institution (1983--87)
2274 UAmsT03M1 13 XIE19960704.0066 was the/a budget director }

**Tuned system answer:**
Rivlin, Alice (Mitchell) 1931 -- Economist, government official; born in Philadelphia. Daughter of a nuclear physicist, she graduated from Bryn Mawr (1952) and earned a Ph.D. from Radcliffe (1958). She became a staff member of the Brookings Institution in Washington, D.C. (1957), was deputy assistant at the Department of Health, Education and Welfare (1966--69), and then became the first director of the Congressional Budget Office (1975--83). She then became director of economic studies at the Brookings Institution (1983--87). In 1993 she was appointed deputy director of the Office of Management and Budget in the Clinton administration. A respected analyst of the U.S. economy, she wrote several studies of economic problems, including Economic Choices (1987) and Caring for the Disabled Elderly: Who Will Pay? (1988).

## *Question 1907 (Who is Alberto Tomba)*

**TREC submission:**

1907 UAmsT03M1 1 APW19981003.0425 0400, tom keyes wrote: &gt;On Thu, Jul 5, 2001 4:18 PM, Gary L. Gray &lt;mailto:gray@engr.psu.edu&gt; &gt;wrote: &gt;&gt;[1] Install the fonts provided to me by Alberto Tomba.
1907 UAmsT03M1 2 APW19981003.0425 $10,000 to the William Moses Kunstler Fund for Racial Justice, established as memorial to the late civil rights lawyer who died on Sept.4 So didAlberto Tomba.
1907 UAmsT03M1 3 APW19981003.0425 1992 Winter Olympics distinguished by the impressive triumphs of Alberto Tomba, Kjetil Aamodt pressure she put on herself that day, RÃ©gine achieved an almost
1907 UAmsT03M1 4 APW19981003.0425 Alberto Tomba hasn&&39;t done squat on the slopes in the one Alpine skier, chances are very good Tomba will be This despite the fact that he&&39;s won only one races
1907 UAmsT03M1 5 APW19981003.0425 Alberto Tomba La Bomba (The Bomb, in Italian) was born into wealthy Bolognese textile family, and never gave up enjoying his privileged life.
1907 UAmsT03M1 6 APW19981003.0425 Alberto Tomba, the big Italian with the even bigger hair, skied with such speed and explosiveness that he was called &quot;La Bomba.&quot; He was the best slalom skier
1907 UAmsT03M1 7 APW19981003.0425 Alberto Tomba was born December 19, 1966, in Castel De Britti, Italy. The same year as the Nagano Olympic games, Tomba won slalom at the World Cup final, which
1907 UAmsT03M1 8 APW19981003.0425 Alberto Tomba was born December 19, 1966, in Castel De Britti on the release of fairytale book written by more recent projects is the Lexus Tomba Tour, giving
1907 UAmsT03M1 9 APW19981003.0425 Alberto Tomba was born December 19, 1966, in Castel De Britti riding, motor cross, and football, Tomba became particularly herein are those of the author and do
1907 UAmsT03M1 10 APW19981003.0425 Amid adoring fans, superstar Alberto Tomba explodes to commanding lead in the World Cup skiing championship Date: 02/20/1995 Reading Level: 8. Publication
1907 UAmsT03M1 11 APW19981003.0425 Amid adoring fans, superstar Alberto Tomba explodes to commanding lead in the World Cup skiing championship (Time International) THICKER, SLOWER TOMBA IS
1907 UAmsT03M1 12 APW19981003.0425 An article in Sports Illustrated reported that skier Alberto Tomba surrounded himself with coach And some clinical psychologists feel those Ph.D.&&39;s in
1907 UAmsT03M1 13 APW19981003.0425 And good luck to all&quot; -- Alberto Tomba, just moments before crushing the field in the A note on sources: David Wallechinsky&&39;s &quot;The Complete Book of the Winter
1907 UAmsT03M1 14 APW19981003.0425 Athlete Bios: Skiing Slalom &amp; GS. Alberto Tomba (ITA) Birthday: 12/19/66 Home Base: Bologna, Italy Skis: Rossignol Boots: Lange Bindings: Rossignol.
1907 UAmsT03M1 15 APW19981003.0425 Baxter has to choose between competing and shaving off the Scottish flag he has died onto his This puts him in class with Alberto Tomba and Katja Seizinger.
1907 UAmsT03M1 16 APW19981003.0425 bio. Alberto Tomba was born December 19, 1966, in Castel De Britti, Italy. Always fond of nature -- especially snow, of course -- Tomba

1907 UAmsT03M1 17 APW19981003.0425 biography. Alberto Tomba was born December 19, 1966, in Castel De Britti, Italy. Always fond of nature -- especially snow, of course
1907 UAmsT03M1 18 APW19981003.0425 emotional moment when Hermann Maier and Alberto Tomba congratulated each Maier was pleased to hear Tomba speaking so compared to him--he has achieved so many
1907 UAmsT03M1 19 APW19981003.0425 FENCING GYMNASTICS BASEBALL CRICKET BASKETBALL WATERPOLO Alberto Tomba, Festina Cycling Team Lizarazu, Vinnie Jones, Warren Barton, Paris University Rugby Club
1907 UAmsT03M1 20 APW19981003.0425 footsteps of such great names as Ingemar Stenmark, Phil Mahre, Alberto Tomba and Buraas&&39; Norwegian teammate Finn Christian Jagge. Buraas, who had died his hair
1907 UAmsT03M1 21 APW19981003.0425 Four years ago at Nagano, Compagnoni wrote herself into the Olympic record that she was considered as Italy&&39;s female answer to the swashbuckling Alberto Tomba.
1907 UAmsT03M1 22 APW19981003.0425 Frenchman Jean-Claude Killy. Some observers already feel the American is worthy successor to Italian slalom ace Alberto Tomba.
1907 UAmsT03M1 23 APW19981003.0425 Giulia was thrilled with the success of the recent Winter World University Games. Alberto Tomba stopped by for look at the Opening Ceremony, in which record
1907 UAmsT03M1 24 APW19981003.0425 He also took silver in the slalom. Discover Alberto Tomba&&39;s biography, Learn more. MEDAL TABLE See prize winners per country: Click here. My video preferences.
1907 UAmsT03M1 25 APW19981003.0425 Here to read more articles on Tomba, Alberto. Search 1Up Info. Search 1Up Info. The Columbia Electronic Encyclopedia Copyright Â© 2003, Columbia University Press.
1907 UAmsT03M1 26 APW19981003.0425 Home, ATHLETES, HEROES &gt; ALBERTO TOMBA, Alberto TOMBA Tomba La Bomba. Charismatic medals. In April 2000 Alberto Tomba received the Olympic Order.
1907 UAmsT03M1 27 APW19981003.0425 I, Alberto Tomba, will finish third in the Super G, second in the Slalom and first in the Giant Slalom.&quot; Tomba reportedly wrote in his autograph book of US
1907 UAmsT03M1 28 APW19981003.0425 I donâ€™t care if youâ€™re trying to generate lead or trying to well-known names in Olympic skiing: Picabo Street, Tommy Moe and Italyâ€™s Alberto Tomba.
1907 UAmsT03M1 29 APW19981003.0425 in Italian ski racing history, and (Sports Illustrated) MASTER OF THE MOUNTAIN Alberto Tomba, the bon vivant of Calgary, won two gold medals (Sports Illustrated
1907 UAmsT03M1 30 APW19981003.0425 In the previous editions, this prize was awarded to Movie and TV Idem, Alex Zanardi, Barbara Fusar Poli and Maurizio Margaglio, Alberto Tomba, Dino Meneghin
1907 UAmsT03M1 31 APW19981003.0425 *). Italian Alpine Skiing sensation Alberto Tomba is 36 todayÂ…. (** He wrote: Â“IÂ’m All Shook UpÂ” as response to the draft notice. **).
1907 UAmsT03M1 32 APW19981003.0425 I thought in days was suspicious. On Thu, Jul 5, 2001 4:18 PM, Gary L. Gray wrote: &gt;[1] Install the fonts provided to me by Alberto Tomba.

1907 UAmsT03M1 33 APW19981003.0425 Laurea degree in Electronics Engineering in 1987 and the Ph.D. degree in iclassics.com Frode: ancora rinviato il processo ad Alberto Tomba [ Translate this

1907 UAmsT03M1 34 APW19981003.0425 Licensed from Columbia University Press. All rights reserved. Related content from on: Alberto Tomba BONUS PIECE: UNLESS YOU ARE TRUE, NEON-BLUE, DYED-IN-THE

1907 UAmsT03M1 35 APW19981003.0425 many other champions, but they&&39;re not Alberto Tomba.&quot; Compagnoni, . at that time when asked about Tomba and the I met him.&quot; Compagnoni took lead of nearly .

1907 UAmsT03M1 36 APW19981003.0425 Money: 52,620 &19 1995 Prize Money: 81,308 &16 1994 Prize Money: 72,021 &17 Career Prize Money: $280,946 Favorite athletes are Alberto Tomba and NBA

1907 UAmsT03M1 37 APW19981003.0425 Network. halls of fame/who&&39;s whoSportsâ€”Halls of Fame/Who&&39;s Whoâ€”R T Alberto Tomba Born: Dec. 19, 1966 Italian alpine skier. winner

1907 UAmsT03M1 38 APW19981003.0425 Norway&&39;s Finn Christian Jagge, leader in the first run with lead of 1/100 on Italy&&39;s Alberto Tomba, lagged at 24th after losing too much time in the second

1907 UAmsT03M1 39 APW19981003.0425 October 1998: &quot;Who knows me knows it&quot; (Alberto Tomba); November 1998: &quot;Stars can frighten October 2001: &quot;First prize: new car; second prize: set of knives

1907 UAmsT03M1 40 APW19981003.0425 of competition, American speed skater Dan Jansen learned that his older sister had died. the story was of young brash Italian by the name of Alberto Tomba.

1907 UAmsT03M1 41 APW19981003.0425 of Hermann merchandise is the &quot;Sturzflug&quot; (Flying Crash), the prize-winning shot list of winners: 1. Ingemar Stenmark, Sweden (86), 2. Alberto Tomba, Italy (50

1907 UAmsT03M1 42 APW19981003.0425 of the world&&39;s ski slopes is the flamboyant slalom specialist Alberto Tomba. Chicago Tribune (February 7, 1992) wrote that Tomba is &quot;perhaps 950 University Ave

1907 UAmsT03M1 43 APW19981003.0425 Only 20 years old, Janica has already entered history, wrote Italian media. Commentator Paolo de Ciesa compared her to Alberto Tomba.

1907 UAmsT03M1 44 APW19981003.0425 peopleAlmanacâ€”Peopleâ€”Biographiesâ€”Sports Personalitiesâ€”R T Alberto Tomba Born: Dec. 19, 1966 Italian alpine skier. winner

1907 UAmsT03M1 45 APW19981003.0425 SCHLADMING, Austria (AP) -- Alberto Tomba, rounding into top form just in time for the Olympics, glided flawlessly to Tomba charged to the lead after the

1907 UAmsT03M1 46 APW19981003.0425 Some journalists wrote afterwards of the disappointment,&quot; Tomba said later. February 20 KC Boutiette â€¢ February 19 Alberto Tomba â€¢ February 18

1907 UAmsT03M1 47 APW19981003.0425 the Lexus Tomba Challenge ended with local ski hero Eric Archer defeating five-time Olympic medalist Alberto Tomba on the This car is the coolest prize have

1907 UAmsT03M1 48 APW19981003.0425 Tomba, Alberto 1966â€“, Italian skier. and silvers for slalom (1992, 1994), Tomba became the Encyclopedia Copyright Â© 1994, 2000, Columbia University Press.

1907 UAmsT03M1 49 APW19981003.0425 to the New Zealander sailor Sir Peter Blake who died under tragic Jean-Claude Killy, Bertrand Piccard, Jesse Owens, Steffi Graf, Alberto Tomba, Indira Gandhi
1907 UAmsT03M1 50 APW19981003.0425 University of Louisville Ladybirds Dance Team View Our Guestbook Sign Our really lack the words to compliment myself today.&quot; Alberto Tomba Copyright Â© 2003
1907 UAmsT03M1 51 APW19981003.0425 Voting Station AmIAnnoying.com Media Kit Alberto Tomba Annoying Not Undecided Please vote to see the next celebrity (Voting Results will appear on Right Sidebar
1907 UAmsT03M1 52 APW19981003.0425 Voting Station The &&39;Mother Angelica&&39; Annoyatorium (Forum) Alberto Tomba Annoying Not Undecided Please vote to see the next celebrity (Voting Results will
1907 UAmsT03M1 53 APW19981003.0425 With their extraordinary appearances and record victories they achieved reputation of 20.12.1987) WCUP VSL Helmut Mayer (AUT) SL Alberto Tomba (ITA) XXVIII.
1907 UAmsT03M1 54 APW19981003.0425 With their extraordinary appearances and record victories they achieved reputation of the top sportsmen and (21.-22.12.1990) WCUP VSL Alberto Tomba (ITA) SL
1907 UAmsT03M1 55 APW19981003.0425 Z. Author: Alberto Tomba, Related Information: Find on Amazon: Alberto Tomba, Send this page to friend Discover Orchids! Get
1907 UAmsT03M1 56 APW19990616.0322 was the/a Olympic gold medalist }

**Tuned system answer:**
3 rel run0 Alberto Tomba La Bomba (The Bomb, in Italian) was born into a wealthy Bolognese textile family, and never gave up enjoying his privileged life.
3 run0 footsteps of such great names as Ingemar Stenmark, Phil Mahre, Alberto Tomba and Buraas' Norwegian teammate Finn Christian Jagge. Buraas, who had died his hair
3 run0 $10,000 to the William Moses Kunstler Fund for Racial Justice, established as a memorial to the late civil rights lawyer who died on Sept.4 So didAlberto Tomba.
3 run0 The record book is now updated to include all records 1996-2002. HISTCARRHISTCARR. I really lack the words to compliment myself today.&quot; - Alberto Tomba.
3 run0 0400, tom keyes wrote: &gt;On Thu, Jul 5, 2001 4:18 PM, Gary L. Gray &lt;mailto:gray@engr.psu.edu&gt; &gt;wrote: &gt;&gt;[1] Install the fonts provided to me by Alberto Tomba.
3 rel run0 ski slopes is the flamboyant slalom specialist Alberto Tomba. watching, he goes faster.&quot; Tomba's personal magnetism Tribune (February 7, 1992) wrote that Tomba
3 rel run0 I thought 2 in 2 days was suspicious. On Thu, Jul 5, 2001 4:18 PM, Gary L. Gray wrote: &gt;[1] Install the fonts provided to me by Alberto Tomba.
3 run0 Some journalists wrote afterwards of the disappointment,&quot; Tomba said later. February 20 - KC Boutiette â€¢ February 19 - Alberto Tomba â€¢ February 18

3 rel run0 Alberto Tomba was born December 19, 1966, in Castel De Britti riding, motor cross, and football, Tomba became particularly herein are those of the author and do
3 run0 Topics | Author Type | Trivia, Authors: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z. Do you want 10,000 Quotations? Author: Alberto Tomba, Click Here.
3 run0 rel of the world's ski slopes is the flamboyant slalom specialist Alberto Tomba. Chicago Tribune (February 7, 1992) wrote that Tomba is &quot;perhaps 950 University Ave
3 run0 Laurea degree in Electronics Engineering in 1987 and the Ph.D. degree in iclassics.com ] Frode: ancora rinviato il processo ad Alberto Tomba - [ Translate this
3 run0 An article in Sports Illustrated reported that skier Alberto Tomba surrounded himself with a coach And some clinical psychologists feel those Ph.D.'s in
3 run0 rel World Cups, three gold and two silvers at the Olympics,twice World Champion: These impressive figures represent Alberto Tomba. He is better known as &quot;La Bomba
3 run0 of Alberto Tomba are available at MaleStars.com. They currently feature over 65,000 Nude Pics, Biographies, Video Clips, Articles, and Movie Reviews of famous
3 run0 Alberto Tomba. Related content from on: Alberto Tomba. BONUS PIECE: UNLESS YOU ARE A TRUE, NEON-BLUE, DYED-IN-THE-GORETex ski groupie (Sports Illustrated).

### *Question 2322 (Who is Ben Hur?)*

**TREC Submission:**
2322 UAmsT03M1 1 NYT20000203.0138 date of birth {1785}
2322 UAmsT03M1 2 XIE19981013.0299 was the/a Nairobi University School of Journalism lecturer }

**Tuned system answer:**
21 rel run0 But there was an earlier, silent version of Ben Hur, also produced by MGM and in charge of the chariot race was B. Reeves &quot;Breezy&quot; Eason, known for his genius
21 rel run0 Ben Hur -1926, Fred Niblo, 1000's of world-famous locations of the greatest movies, top TV shows, film stars, soap operas, directors, bestselling writers and
21 rel run0 Famous Location Ben Hur - 1959. 'Ben Hur' Oasis Folliano Rome Lazio Italy, Famous Location Ben Hur - 1959. 'Ben Hur' Town 1 Fiuggi Rome Lazio Italy,
21 rel run0 Return to home, Ben-Hur (1959). If we are not, you will sink with this ship, chained to your oar.&quot; Ben Hur refuses the offer unless it means his freedom.

# References

[1] Agichtein E., Gravano L. Snowball: Extracting Relations from Large Plain-Text Collections, Proceedings of the Fifth ACM International Conference on Digital Libraries .2000.

[2] Argamon S. Koppel M. and Shimoni A. R. Automatically Categorizing texts by Author Gender. Literary and Linguistic Computing, (in the press 2003)

[3] Banko M. Brill E. and Duamis. S. An analysis of the AskMSR Question-Answering System. Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing. 2002.

[4] Banko M. Brill E. Duamis. S. Lin J. and Ng A. Web Questions Answering: Is more Always better?

[5] Bing Liu, et al. Mining Topic-Specific Concepts and Definitions on the Web. 2003.

[6] Blair-Goldensohn S. McKeown K.R., Schlaikjer A. A Hybrid Approach for Answering Definitional Questions.

[7] Chu-Carroll J. and Prager J. Use of WordNet Hypernyms for Answering What-Is Questions. TREC-2001. 2001.

[8] Cohen W. and Singer Y. Context Sensitive Learning Methods. Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval. 1996.

[9] Dawn M.T. and Voorhees E.M. Building a Question Answering Test Collection.

[10] De Rijke M. et al. The University of Amsterdam at TREC 2003. (To appear at TREC-2003)

[11] Ghanem M., Guo Y., Lodhi H. .Automatic Scientific Text Classification Using Local Patterns: KDD CUP 2002. SIGKDD Explorations. Volume 4, Issue 2. 2002

[12] Graham L. and Metaxas P. T. “Of course It’s True; I Saw it on The Internet!”: Critical Thinking in the Internet Era. Communications of the ACM May 2003.

[13] Hull D. Using Statistical Testing in the Evaluation of Retrieval Experiments.

[14] Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of ECML-98, 10th European Conference on Machine Learning, 1998.

[15] Kessler B. Nunberg G. and Schutze H. Automatic Detection of Text Genre. Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics, 1997.

[16] Knight K., Marcu D. Summarization Beyond Sentence Extraction: Summarization Beyond Sentence Extraction, Artificial Intelligence. 2002.

[17] Lin C.Y. The Effectiveness of Dictionary and Web based Answer Reranking. In *Proceedings of* the 19th International Conference on Computational Linguistics (COLING 2002), 2002

[18] Mani I. Automatic Summarization. John Benjamins Pub Co. 2003.

[19] Mani I. Summarization Evaluation: An Overview. 2001.

[20] Manning D. and Schutze H. Foundations of Statistical Natural Language. MIT Press, Cambridge MA, 2000.

[21] NIST. 2003. TREC 2003 Question Answering Track Guidelines.

[22] Sebastiany F. Machine Learning in Automated Text Categorization. ACM Computing Surveys, 2002

[23] Voorhees E. M., Tice D.M. The TREC-8 Question Answering Evaluation. TREC-8. 1999.

[24] Voorhees E.M. Evaluating Answers to Definition Questions.

[25] Voorhees E.M. Overview Of the TREC-2001. TREC-2001.2001

[26] Voorhees E.M. Overview Of the TREC 2002 Question Answering Track, TREC-2002. 2002

[27] Voorhees E.M. Overview Of the TREC 2003 Question Answering Track (Draft)

[28] Voorhees E.M. Overview of TREC 2003 (Draft).